2020

Nyku: A Social Robot for Children With Autism Spectrum Disorders

Dan Stephan Stoianovici
University of Denver

Nyku : A Social Robot for Children with Autism Spectrum Disorders

---

A Thesis

Presented to

the Faculty of the Daniel Felix Ritchie School of

Engineering and Computer Science

University of Denver

In Partial Fulfillment

of the Requirements for the Degree

Master of Science

---

by

Dan Stoianovici

August 2020

Advisor: Dr. Mohammad H. Mahoor

Author: Dan Stoianovici
Title: Nyku: A Social Robot for Children with Autism Spectrum Disorders
Advisor: Dr. Mohammad H. Mahoor
Degree Date: August 2020

Abstract

The continued growth of Autism Spectrum Disorders (ASD) around the world has spurred a growth in new therapeutic methods to increase the positive outcomes of an ASD diagnosis. It has been agreed that the early detection and intervention of ASD disorders leads to greatly increased positive outcomes for individuals living with the disorders. Among these new therapeutic methods, Robot-Assisted Therapy (RAT) has become a hot area of study. Recent works have shown that high functioning ASD children have an affinity for interacting with robots versus humans. It is proposed that this is due to a less complex set of communication modes present in a robotic system as opposed to the complex non-verbal communications present in human to human interactions. As such, the Computer Vision and Robotics Lab at the University of Denver has embarked on developing a social robot for children with ASD.

This thesis presents the design of this social robot; Nyku (Figure 1). It begins with an investigation of what the needs of ASD children are, what existing therapies help with, and what, if any, roles a robot can play in these

treatment plans. From the literature examined, it is clear that robots designed specifically for ASD children have a core set of goals, despite the varied nature of the disorder's spectrum. These goals aim to reduce the stress of non-verbal communications that may occur during standard therapies, as well as providing capabilities to reinforce typical areas of weakness in an ASD persons social repertoire, such as posture mimicry and eye contact. A goal of this thesis is to show the methodology behind arriving at these design goals so that future designers may follow and improve upon them.

Figure 1: Nyku

Nyku's hardware and software design requirements draw from this foundation. Using this "needs first" design methodology allows for informed design such that the final product is actually useful to the ASD population. In this work, the information collected is used to design the mechanical components of Nyku. These elements consist of Nyku's Body, Neck & Head, and Omni-wheel base. As with all robots, the mechanical needs then spawn electronics requirements, which are, in turn, presented. In order to tie these systems together, the control architecture is coded. Notably, this thesis results in a novel kinematic model of a spherical manipulation system present in the Omni-wheel Base. This solution is then presented in detail, along with the testing conducted to ensure the model's accuracy.

To complete the thesis, overall progress on Nyku is highlighted alongside suggestions for a continuation of the work. Here, the engineering work is compared against the design goals which it tries to fulfill in an effort to ensure that the work has stayed on track. In continuation, this examination maps out future steps needed to optimize the engineering work on Nyku for reliable performance during therapeutic sessions. Finally, a therapeutic plan is proposed given the hardware capabilities of Nyku and the needs of ASD children against the background of modern therapeutic methods.

Acknowledgements

I would like to express my gratitude to the Ritchie School of Engineering and Computer Science for supporting me and providing me with the opportunities to pursue my dreams of becoming a roboticist. A community where I built my most valuable friendships and discovered my passion.

And to Dr. Mahoor, my advisor, for believing in my abilities even when they weren't apparent. Thank you for the opportunity to better myself and the guiding hand when the path was not clear. Working in your lab has been a pleasure and is a badge I wear with honor.

During my time at DU, I was fortunate to have many mentors. I would like to thank Josh Lane and Hojjat Abdollahi for having the patience to bring me up to speed in the world of robotics with a smile and a joke.

Importantly, I would like to thank my mother and father for pushing me in my pursuits and always allowing time to play. To my mother, who has always encouraged me to speak my mind, for better or worse. And to my father, who has always shared his technical expertise and who inspired the solution for the inverse kinematics controlling the robot's base. Both of you have always instilled a love of learning, problem solving, and teaching.

Table of Contents

List of Figures

Chapter 1

Introduction

1.1 Background and Significance

Autism Spectrum Disorders (ASD) are a set of neuro-developmental disorders that afflict many children in the US and abroad. These diseases are characterized by many comorbid signs and symptoms which present themselves in a combination unique to the person living with them. Persons living with ASD can be perceived as having strange body posture and facial expression while having a difficult time reading social ques. In children, autism presents itself at a young age with a delay in the ability to speak or to comprehend speech. Often these symptoms are interpreted by others as anti-social behavior. With therapy, the outcomes of people living with ASD can be greatly improved as they learn to navigate their surroundings from an early age.

As of 2018, the US Center for Disease Control (CDC) has reported a rise in the rates of autism from 1 in 68 children age 8 to 1 in 59 (1.5% to 1.7%)

[29]. In 2012, a congressional hearing was held to discuss the rise in autism rates, which at that time was 1 in 88 children, regarding the lack of research funding for both the causes of Autism spectrum disorders and the therapies for those afflicted federal support [1]. And in 2004, the CDC also rolled out its "Learn the signs, Act Early" (LTSAE) initiative in an attempt to screen all at risk children for ASD before 36 months and enrolling them in community based support systems by 48 months. Despite these efforts, Autism rates have grown 150% from 2000 to 2018. According to the World Population Review, the rate of Autism in the United States are currently 1 in 45 children age 8 [41]. Although this may be attributed an increase in screening and awareness, the rates of ASD disorders across the globe have followed this trend. As seen in Figure 1.1, the United States is third in the world behind Hong Kong and South Korea for rates of autism. Despite being declared a pubic health crisis by the CDC and the world wide concern, there has been no breakthrough research on the cause of Autism Spectrum Disorders. As such, there is a large body of research on therapies that can improve the outcomes for people living with ASD.

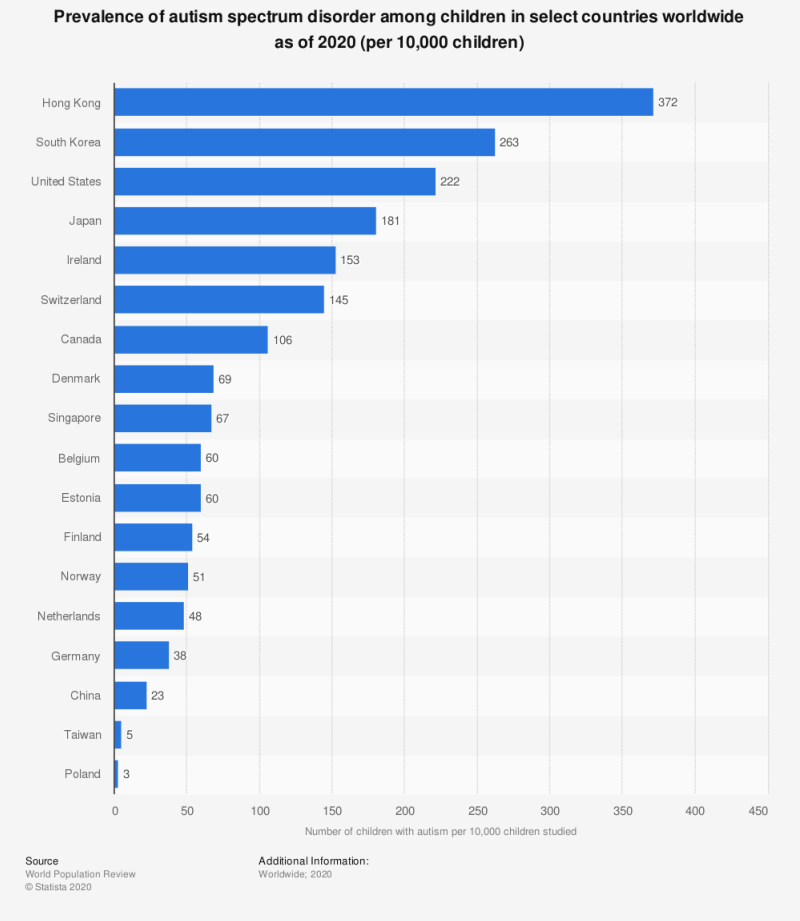


Figure 1.1: World Autism Rates from Statista According to the World Population Review April 6th 2020 [41]

It is agreed that early detection and intervention of ASD leads to the best outcomes for these people. This intervention can take many forms including various forms of therapy, community support groups for those living with ASD and their caretakers as well as dietary plans. Anyone who is a care taker, or anyone living independently with ASD, knows that these resources

and therapies must be combined into a comprehensive plan. These plans depend heavily on the needs of the person living with ASD which in turn are based on the severity of the disorder. These plans often include time spent with behavior and speech therapists as well a psychologists and psychiatrists if the person's ASD has other comorbid symptoms. Because of the wide variety of needs that an ASD child may have, the treatment plans are just as diverse as the disorder itself.

Recently, with calls for an increase in Human Robot Interaction (HRI) studies, there has been an elevated interest in social robotics concerning the treatment of ASD children, specifically Robotics Assisted Therapy (RAT) for ASD children. These early studies have shown encouraging results demonstrating positive outcomes for the ASD children. In addition to this, the use of robots in therapy for children with ASD can help alleviate the well documented fatigue that occurs in ASD caretakers [10].

As a burgeoning field, RAT for ASD children is showing exciting early results, however the development of the research is limited by some factors. First, the robots used in the RAT studies are, more often than not, very costly. Most studies are conducted with NAO robots, which cost upwards of $10k for a new model, or they are purpose built robotic systems not made widely available to other research institutions. An institution interested in RAT for

ASD children must therefor invest heavily in either off the shelf hardware or development costs. The issue with off the shelf hardware is that it is not built with the needs of ASD children in mind. And the issue with creating a robot for RAT is development time. The second main issue is that the robots are not designed with the all needs of ASD children in mind. For example, NAO, is marketed as a humanoid robot development platform and includes some features that may hinder its effectiveness as a therapeutic tool. Purpose built robots, on the other hand, are so specific that they only lend themselves to one specific experiment. While there are commercial robots designed for ASD children such as Keepon, their ability to accommodate a wide range of research is limited.

Given the increased interest in STEM education and the onset of the Maker movement, there have been a number of companies producing open source hardware for engineering development. More and more products encouraging the development of robotics are hitting the market at consumer targeted prices. Powerful micro computers boasting 4GB of RAM, like the Raspberry Pi 4B+, now cost $50, and closed loop control servos can now be bought for as little as $5. Given this abundance of available robotics hardware, purpose built systems can be made at a reasonable price point, and the plans

made available to the public. This creates the perfect environment for the development of ASD specific robots.

In order to advance the field of RAT and provide a platform for future research, the Robotics and Vision laboratory at the University of Denver has engaged in the development of a social robot for children with autism called Nyku. In this thesis, the development process for Nyku is detailed. Chapter 2 begins with an examination of the needs of ASD children, the robots trying to meet these needs, and the studies conducted in order to measure the efficacy of the RAT technique. Using this information in Chapter 3, the requirements of the robotic system are determined and the resulting design is presented. From this design, novel kinematics arise for the control of a sphere and are subsequently presented in Chapter 4. To conclude with Chapter 5, the current progress of Nyku is discussed and future research steps are highlighted.

1.2 Problem Statement

The lack of research in the field of Human Robot Interaction (HRI) concerning the efficacy of Robot Assisted Therapy (RAT) is caused by the expense of the equipment needed to conduct the research and the disconnect between Psychologists and the Engineers developing the robots. Due to this lack of research, robot assisted therapy is not commonly used in the treatment

plans of ASD children, despite promising preliminary results. The recent surge in open source robotics hardware allows for the development of inexpensive robots which can be designed with very specific goals in mind. In order to expand the research horizons of RAT, inexpensive robots that meet the needs of ASD children must be designed and made easily available to those interested in conducting HRI and RAT studies.

1.3 Thesis Organization

This thesis continues in Chapter 2, where the background research used to develop Nyku is presented. Here the fields of HRI and RAT are studied in order to gain insight on where the work currently is and how it is being applied to ASD children. In Chapter 3, this information is used to develop engineering requirements for Nyku which are then extrapolated into the final design which is also detailed in this chapter. As a result of this design, novel kinematics are necessary for the control of the system which are presented in Chapter 4. This thesis concludes with Chapter 5, where the design process is reviewed along with the contributions of the work, necessary next steps to continue the research, a discussion of the work, and some concluding thoughts on the process.

Chapter 2

Background Research

2.1 Autism Spectrum Disorders

Autism, from the Greek root "autos", is a disorder of the insulated self. The term appears in 1910 when Eugene Bleuler, a Swiss scientist credited with coining the term schizophrenia, used the word autism to describe the insular tendencies of schizophrenic adults. In the early 1940's, the Autism we are familiar with today was described by Leo Kanner and Hans Asperger independently and contemporaneously, although there is some dispute on if this was true co-discovery. In Kanner's and Asperger's descriptions, Autism becomes a syndrome of its own as opposed to a symptom of another condition [24]. Recently, it has been unearthed that Kiev-based child psychiatrist, Grunya Efi movna Sukhareva, published a description of "schizoid (eccentric) psychopathy" in 1925 that starkly resembles the modern DSM-5 criteria for

Autism diagnosis. She later, in 1950, changed the terminology to "autistic psychopathy" in her Russian language papers [34].

Today, Autism, or Autism Spectrum Disorder (ASD), refers to broad range of developmental disorders characterized by impairments in imagination, interpretation of social ques, verbal & non-verbal communication, and restricted repetitive behaviors [16]. Although a diagnosis requires a combination of social deficits and repetitive behaviors, the intensity of individual symptoms does not preclude the diagnoses. While an ASD person may be greatly impacted by their social deficits, their repetitive behaviors may not interfere with their functioning as much. Others may exhibit less severe social impairments but be hindered heavily by their stringent repetitive behaviors. Because of this variability in symptoms, autism is considered a spectrum disorder.

Although there are many comorbid symptoms of ASD that can be studied, we focus on the non-verbal communicative (NVC) deficits that are typical of the population, specifically in children. According to the DSM-5, the deficits in NVC manifest in "reduced, or atypical use of eye contact (relative to cultural norms), gestures, facial expressions, body orientation, or speech intonation". One of the earlier expressions of autism is an impairment in joint attention, where the child has difficulties following someones eye gaze or pointing gesture,

and does not reciprocate the actions. It is also reported that, despite individuals eventually learning a few functional gestures, their repertoire of use and understanding of these gestures is limited in comparison to neurotypical (NT) populations [16]. In verbal ASD adults, this can results in odd, "wooden", or exaggerated body language.

It is hypothesized that these limitations in gesture and body language are due to a difference in how the ASD population processes body features. There is evidence showing that both the body and face are interpreted using configural processing in neurotypical adults [42]. Processing here refers to the mechanism used to interpret individual features or stimuli, such as facial or body features. Configural processing refers to understanding the whole by comparing individual features, such as looking at the eyes, ears, nose, and mouth individually to acquire meaning from the whole. In holistic processing meaning is derived from observing all of the features together, as a whole. Relational processing refers to the understanding of features in the context of one another. In their article, Reed et al. conduct a body inversion study on neurotypical adults as well as ASD adults to see whether or not the same trend is present in body processing as in face processing [42]. Inversion studies attempt to determine if the subject uses configural processing to interpret information from certain stimuli. These studies are usually conducted by pre-

senting the stimuli right side up, and asking the participant to recognize the stimuli when it is inverted. The presence of a large inversion effect indicates that the participant is interpreting the stimuli using configural processing. It has been shown that ASD individuals do not present a significant inversion effect when presented with face stimuli [15], indicating a deficit in configural processing for that specific stimuli. Reed et al. show that, in their high functioning adult ASD population, there is little inversion effect, supporting that there is a deficit in configural processing when interpreting body posture as well. Because of the small population in this study, the findings alone are inconclusive, however they are supported by deficits in biological motion as indicated by the point-light animation studies done by Blake et al. [8]. Blake also notes that the severity of a child's ASD is positively correlated to the prediction of biological motion.

Studies involving processing mechanisms of ASD populations are not limited to inversion effect tests. In their 2003 paper, Joseph and Tanaka conduct a study on face processing using face composites. Faces were shown whole, and then the eyes, nose, and mouth were rearranged on a digital screen very quickly. Typically developing (TD) children and ASD children were tested and it was found that the TD children had a harder time recognizing reorganized faces, showing that configural processing is in effect, while the ASD children

showed no preference. It should be noted that no group showed a bias for nose based identification [25]. It was also shown that the ASD group showed a significant affinity for the mouth based identifications and a deficit in eye based identifications. The eye aversion tendency is well known, and Reed et al. suggest that body posture may be a better method of communication with ASD children as they do not have to maintain eye contact.

2.2 Posture and Mimicry as a Mode of Communication

As it has been proposed that posture and motion may be better methods of communicating with ASD populations, we examine what posture conveys. Posture has been analyzed as a mode of emotional communication since Darwin's *The expression of the emotions in man and animals* where he proclaimed that emotions and emotional expression evolve in tandem with biology [14]. Furthermore, bodily articulation is accepted as a diagnostic indicator in acting and dance therapies [17]. By observing the way that the patient moves and holds themselves during these somatic exercises, the therapist can gain insight to the patients state of mind. Therapists were able to gauge the state of mind of the patient from how they held their body in the performance, noting "open" and "closed" postures. In classical ballet, the entire story is mimed by the performers to portray a complex narrative. An analysis of ballet dancers

found that the performers assumed round postures when representing warm and likeable characters, while antagonistic characters were depicted by more angular postures [3]. In their examination of ballet, Aronoff et al. propose that the geometrical properties of dancer's poses signal the emotional undertones of the performance. More over it has been shown that even assuming a posture alone can elicit the emotional response.A 1999 Journal paper by Flack et al. ran an experiment where subjects were asked to assume certain poses correlated with certain emotions: Anger, Happiness, Fear, & Sadness [20]. They found that, without being told which pose they were assuming, the subjects could accurately predict the emotion that the posture was meant to exhibit.

While dance & acting therapies and ballet consist of pose, they are mainly kinetic expressions. However there is also evidence that static posture is an effective means of communicating emotion [12], but there have not been many studies that explain what body posture features are most important in this communication. In his 2004 journal paper, Mark Coulson examines the role individual body segments contribute to the perception of emotion from static posture. Coulson hypothesized that by presenting a relatively large sample population (61 people: 36 female, 27 male) with 176 faceless CGI mannequins, each statically posed in one of the 6 basic emotions, anger, dis-

gust, fear, happiness, sadness, and surprise, he could deduce which limbs or combination thereof contributed to the certain emotions. On that front, his research was inconclusive as no limbs were found to be redundant in communicating each emotion; he posits that there may be an unidentified perceptual variable that is not being measured that leads to these results. The research, however, validates other important principles. Importantly, Coulson's findings show that emotion recognition from body posture is on par with recognition from face and voice for predictive accuracy. He also notes that a forward facing posture was the most accurate view for the prediction of emotion. Although the front facing position does not give the best angle for viewing all body angles, Coulson suggests that it may be effective because of the interpersonal meaning of the pose.

Coulson's claims on the interpersonal meaning of forward facing pose are widely supported in literature. In a 1980 article published in the *Journal of Non-Verbal Communication*, Deborah Trout and Howard Rosenfeld report their findings on how posture and limb mimicry affect the perceived rapport between a psychiatrist and patient. In their study, participants were asked to watch a staged therapy interaction videos without sound and facial ques. The participants were then asked to rate the rapport between the therapist and patient. The staged therapy sessions are varied in body posture and

limb mimicry, or congruence. It was found that, in the absence of other non-verbal ques that a forward body posture indicated to the observer that there was a high level of rapport between the therapist and patient. There was not a significant change in interpreted rapport when the limb posture was in-congruent as the data was similar to that of the congruent limb posture [48].

Although Trout and Rosenfeld could not attribute limb congruence, or mimicry, to positive psychotherapeutic rapport, the importance of somatic mimicry as a communicative tool has been the subject of many studies. In a 2 part University of Victoria study on motor mimicry as communicative act, Bavelas et al. demonstrated that mimicry is a result of a trusting relationship between two parties [6]. In the first part of their experiment, the researchers stage a situation where one of the study proctors appears to damage their finger while setting up the experiment. Unbeknownst to the subject, the amount of eye contact during the interaction is varied, and via a camera the researchers observe and quantify how the subject mimics the pain of the study proctor. In the second part of the study, videos of the previous experiment were shown to a new set of subjects who were tasked with decoding the emotional response of the subject in the previous experiment. From these two experiments the researchers found that while an interpersonal connection between the two parties

mimicking strengthens the mimicry response, the mimicry response is not fully dependent on this connection, which was emulated with eye contact. They did note that the mimicry tended to happen during eye contact, as the subject knew that the proctor could see the action. In their own words, Bavelas et al. claim that mimicry is an act of communication indicating that "I am like you, I feel as you do".

2.3 Why Robots?

Robots have been widely used in the medical field; there are surgical assistive robots that allow physicians to perform surgeries remotely, be it in India separating twins, or on the battlefield providing the best medical care to victims of war. They have also been used physical therapy for stroke patients to retrain muscles in an anatomically correct way [31] [21]. At research done at the University of Denver Computer Vision and Robotics lab, a robot Ryan was developed for elderly patients suffering from Alzheimer's and dementia [2]. This robot was developed to assist with the care of these patients by reminding them to take their medications, playing memory games, and initiating dialogues about life experiences. The pilot study, conducted on 6 subjects, showed that the robot established a rapport with the patience and was well accepted. A unique feature of Ryan is its rear projection face, which

allows for accurate animation of facial expression. In another pilot study, TD and ASD subjects were asked to identify the facial expressions that Ryan was animating with varying intensity. Both TD and ASD subjects were able to accurately identify the facial expressions, showing no deficits between the groups [4]. There was, however, confusion between "disgust" and "fear" emotions. This confusion was improved as the intensity of the facial expression was increased.

An interesting study by Marian Banks compared the effectiveness of living dogs versus robotic dogs (Sony AIBO) in reducing loneliness in elderly people living in a nursing home. Banks et al. found that the robotic dog was comparably effective at reducing loneliness, even when the robot dog's facial recognition software and full mobility functions were not used [5].

The goal of most social robotics designed for children with autism has been to assist in the development of normal play [43]. Play has been shown to aid in the development of appropriate social behavior, complex problem solving, and the ability to draw meaning from spoken language [49]; areas in which people with ASD need improvement.

Research has suggested that ASD children prefer social interactions with robotic entities. A 2013 survey paper on social robotics for children with autism cites evidence that because of its simplified nature, children with

autism are more likely to engage socially with a robot than they would be with a human [9]. By reducing the complexity of physical social features in the design of these robots, the children are more easily able to initiate social interactions with the robot than they can with a human being. Since robots are precisely controlled, they are able to gradually modulate which social cues are being presented. In a social robot for children with autism the robot can remove as much non-verbal and verbal communication as necessary so that the child is not intimidated by the interaction [43]. Using this incremental approach to communication allows the children to become comfortable with communication features, such as eyes, without having to take in the complexity all at once. The goal is to generalize this comfort to interactions with humans and elicit healthy play.

Keepon, a social robot developed by Hideki Kozima at the Japanese National Institute of Information and Communications Technology, was used in normal play at a school for children with developmental disorders (DD) to validate its design [30]. In the school, Keepon was presented as one of the toys for the children to play with, it was left in the play room ground like the other toys. During the year and half Keepon spent in the school, it was involved in over 500 interactions with children with autism and other developmental disorders. Among other positive outcomes, a case study from the trials found

that a young girl with more severe DD was found to increase her engagement with Keepon over 18 months, eventually engaging in triadic imitation play with Keepon, her caregiver, and her parent.

In another social robotic study done at the University of Southern California, researchers studied the effects of interactions with the robot on the quantity of social interactions in ASD children [19]. Two robots were tested, one was a mobile bubble blowing robot, meant to emulate the widely used technique in standard play therapy. The other robot was a humanoid robot meant to show exaggerated gesture or posture and asks the child to follow in a Simon Says type game. Fiel-Siefer et al. found that, when using a combination of both robots, the children's verbal utterances increased from 39.4 to 48.4. This increase is in total utterances to the robot, caregiver, and parent.

Despite the successful nature of Robot-Assisted Therapy in children with autism spectrum disorders, and the importance of posture in non-verbal communication, there have not been many studies conducted on the effects of robotic posture mimicry on communications with ASD children. Most studies in this area are concerned with the validation of the emotion projected by the posture, such as in Erden's paper where he extends Coulson's work on limb angle in relation to emotional posture [18]. In this paper, Erden emulates human postures using the NAO robot as a human proxy. Although Erden

does find that the specified postures elicit the intended emotion with a high recognition rate, his study focuses on creating robotic posture from humanoid form and does not concern the impacts of emotional posture on the perception of the robot and its impact on the ASD population. Because of the humanoid form of NAO, it is easy to transpose the emotional postures of people, but it also adds complexity to the non-verbal communication of the robot, which is undesirable in ASD populations. There have been studies using NAO to study posture mimicry and its effect on ASD children. In one paper, NAO is used as a posture model for children with ASD. In this work, Greczek et al. demonstrate a pose on NAO and ask the ASD child to follow the posture [23]. The team uses preset and graded feedback to correct the child's pose. Although the graded cueing feedback was not fully used because the children could easily assume NAO's pose, it is important to note that this is one of the reasons that robots are used; because of the ability to give consistent and appropriate feedback, without deviation that may disturb the process. A paper published in the International Journal of Social Robotics, by Isao Fujimoto et al. outlines a very similar experiment where a humanoid robot is used to instruct an ASD child during an imitation game with modulated feedback on the quality of imitation. However the majority of this paper is written about how the imitation motion was planned and evaluated [22]. In another study,

NAO was simply used to mimic the pose that an ASD child assumed, however their work does not focus heavily on how this affects the child [33].

These works have been done on humanoid robots, whose degree of freedom (DOF) similarity aids in mimicking posture and conveying bodily emotions as the observed pose can be more analogously mapped to the robot. There has been research done on mapping the posture of human pose to more simple robots with fewer degrees of freedom. In one note-able paper, Roshni Kaushik and Amy LaViers experiment with a virtual robot, Broombot, which has 2 articulated degrees of freedom; tilting about both y and x axes [27]. In this work, Kaushik and LaViers map human motion capture movements to the 2 DOF Broombot. They separate the motion capture videos by which limb expresses the most motion during the clip, and then the motion is mapped to the Broombot for that most mobile limb. For example, some videos contain images of people moving their right arm a lot, so in these videos, Broombot's motion is mapped from the body markers on the right arm. This mapping as done for the right & left arms, right & left legs, as well as the spine. The subjects were then presented with a skeletal representation of the motion capture videos alongside 2 Broombots each imitating the video, but with different mappings. Importantly, a significant group of subjects said that the Broombot best imitated the human skeletal representation when the Broombots motion

was mapped from the spinal segment. Although this study was not ran on children with autism spectrum disorders, it does provide a baseline for neurotypical preference in low degree of freedom imitation systems.

In RAT studies done by members of the DU Computer Vision and Robotics lab (DUCV), robots have been shown to improve social skills in ASD subjects. In a 2015 study, researchers in the lab used a NAO robot to teach seven children prototypic facial expressions from pictures [36]. In this experiment, NAO's expressionless face provided the neutral affect that allowed the discussion of the facial expression in the pictures present. Without overwhelming the children with a therapists own facial expressions the children were able to focus on learning the expressions in the picture, resulting in a statistically significant improvement in facial expression recognition. In an earlier study from the lab, Mavadi et al. demonstrated that TD children adapt their gaze patterns depending on the context of the situation, whereas ASD children's gaze patterns are predictable using a Variable-order Markov Model (VMM) of order one for varying eye gaze situations [35]. This lays an important frame work for HRI as it demonstrates the specific needs of ASD children. Using this information as background the lab was able to produce a study validating the experimental design of a robot based therapeutic protocol, again with NAO [37]. In this study, the team lays out the procedure for a RAT

study, and found that when conducting it the ASD participants were able to extend their learned behavior from NAO into human to human interactions afterwards. Of course, with all RAT studies at this stage, there is a limited sample size. However, these preliminary results are very promising. In another study, the team, in conjunction with Dr. Anibal Gutierrez from the University of Miami, used another humanoid robot, Zeno, to conduct HRI studies with Applied Behavioral Analysis techniques [46]. In this study, they measured the response of the children to certain reinforcements and used the children's best reinforcements to teach emotion recognition through a picture matching game. Again, these studies suffered from a small sample size. Building off of this research, the lab used Zeno to conduct a study measuring the impact of gestures on emotion recognition in both TD and ASD children [45], where it was found that the combination of facial expression and gesture lead to an increase in the emotion recognition in both groups, given that the emotion has associated gestures.

Due to its interest in the field, the DUCV lab has produced another social robot, eBear [51]. This robot is focused on animatronic facial expression with 10 DOF and an LCD mouth with synthesized speech. This robot was used in a pilot study measuring the impact of the design on increasing the mood of elderly people with moderate depression. It was found that using

eBear in 45 minute sessions as a companion helped increase the mood of the elderly subjects as measured with the Face Scale and the Geriatric Depression Scale scoring systems [26].

As with any new field, there are initial shortcoming in the research that must be filled. In the case of RAT studies, the gaps are common across the board. First, most of these studies have a very small sample size. This limits the extrapolation of the results onto larger populations. One cannot say that their robotic therapeutic model help all ASD children if their study only measures 10 children over 6 sessions. Although some studies, like those done with Keepon, have attempted longitudinal data collection, they have not been able to increase their number of participants. This is due to the inherent difficulty of keeping test subjects committed to the experiment for multiple years. A second issue with these studies is the qualitative and unstandardized methods of collecting data from the test subjects. This can lead to skewed data collection resulting in poor interpretation. Along side this, the nature of the research tends to attract technically minded researchers such as engineers and computer scientists, who may not be qualified to establish good therapeutic practices. Although collaboration between fields is desired and exists, such as in the DU research done with Dr. Gutierrez from U. Miami, the research is still led by those studying technical fields. Furthermore, there

is a complete gap in research concerning how posture mimicry in robots affects therapeutic rapport and efficacy. Although, this has been studied in human to human therapies, work concerning posture and robotics has been mostly simple simon-says exaggerated poses more than natural posture. There have been no studies where natural posture mimicry has been a blind independent variable.

Chapter 3

Designing a Social Robot

3.1 Design Goals and Methodology

Nyku is a social robot designed for children with ASD. The goals of this design are to build a robot capable of mimicking human body posture while serving as a remote avatar for a therapist. Having examined previous works in the field of human-robotic interaction and the needs of children with ASD, Nyku is developed in order to meet a specific set of criteria:

1. The robot must appeal to children with Autism Spectrum Disorders.
2. The robot must not elicit anxious behaviors from the user.
3. The robot must be able to communicate with simple non-verbal gestures.
4. The robot must be able to mimic human body posture.
5. The robot must be able to serve as a remote therapist.

6. The robot must be easy to manufacture such that it can be used as a cost effective HRI research tool.

In order to appeal to the children with ASD, the robot is designed as a simplified penguin; this is done to avoid Mori's Uncanny Valley [39]. Although there are concerns of acceptance when designing any social robot, these concerns are minimized by designing Nyku with a zoomorphic form [9]. Another large concern of the robot is to avoid the over stimulus of the child with ASD. This over stimulus can occur if there are more methods of NVC conveyed than can be interpreted by the ASD child. This over stimulus can overwhelm the child and reduce the therapeutic benefits of the robot [30]. To ameliorate this problem, Nyku has been designed with simplified NVC in mind. The two main methods of NVC come from the face, through expression, and from the body, through posture.

In order to avoid eliciting the anxious behaviors in the children, which may hinder communication between the robot and the child, we propose to simplify the communicative possibilities of the robot. To simplify the face based NVCs, Nyku is designed without a mouth, leaving only eye expression which can be coded to display the NVC. In Fig 2.1 we see the first version of Nyku with the eyes display on. By omitting the mouth, we remove a large factor of communication from the face which ASD children rely on [25]. This aim

is to allow the ASD child to take in the eyes alone, without distraction. This also clears some communication bandwidth so that the child can effectively receive verbal communications from the therapist.

Figure 3.1: Nyku V1; Dressed with Eyes Open

To simplify the posture NVC from Nyku, we model the penguin body with three links connected with two spherical joints: the inertial base link, torso link, and head link. The torso link also contains two independently actuated "flippers" to achieve anatomical resemblance to a real penguin. These serve a dual purpose of allowing Nyku to more easily express more animated

emotions such as excitement or surprise. Each of the body links has 3 DOF with a range capable of more that the human body, Figure 3.2, so that it can exaggerate postures if necessary to dramatically convey emotions to the children. $L1$ represents Nyku's torso link, and $L2$ represents the head link, the inertial link is omitted from the diagram for simplicity. The position of these two links controls Nyku's posture.

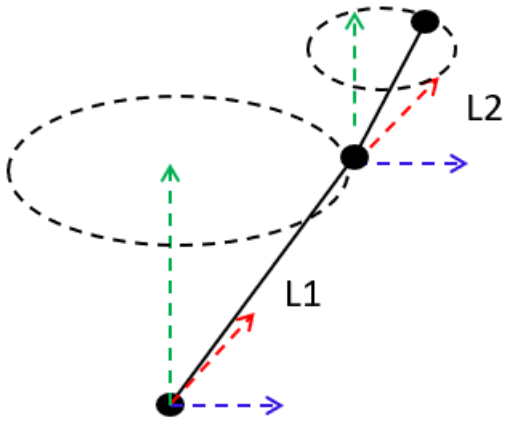


Figure 3.2: 3D Robot Work-space

In order to act as a remote therapist, Nyku needs to be able to relay communications between therapist and patient. We achieve this by adding a camera, speakers, and a microphone. Using these tools, the therapist can hear and see the child, and the child can hear the therapist. While acting as a remote therapist, Nyku will also have to maintain eye contact and posture

mimicry. This is achieved through the internal camera which can center the patients gaze and an external camera that allows for the use of deep neural networks to detect the patients body posture. With these tools, the therapist can focus on the patient and not the control of the robot.

The design methodology of Nyku has been to simplify mechanical construction and use open source electronics in order to make the robot easy to reproduce in house. Nyku can currently be produced using only a large format 3D printer and a laser cutter. Nyku's other components can be sourced from common robotics hobby retailers. Full detail of hardware components can be found in Appendix A. Nyku's control code and architecture is made publicly available in the GitHub repositories in Appendix A as well. Given the ready availability of these components and their relatively low cost, it is feasible that Nyku may be constructed at any facility interested in HRI, making it a strong tool for furthering research in the field without being cost prohibitive.

3.2 Mechanical Design Overview

Using the overall design goals, technical specifications are now considered so that these goals can be achieved. The overall mechanical design of the robot is presented here. A full components list can be found in Appendix A, but will be omitted here, except when necessary, for brevity. The system con-

sists of two main parts: Nyku's body and the Omni-Wheel Base upon which it sits. The body contains links $L1$ and $L2$, as shown by Figure 3.2, the torso and head, which remain untethered from the Omni-Wheel base allowing for full mobility of the links. The two link system mandates that there are two joints. These roles are fulfilled by the Omni-wheel Base, which connects the body ($L1$&$L2$) to the inertial ground link, and the neck, which connects the torso ($L1$) to the head ($L2$). In this section we present the mechanical design of these components.

3.2.1 Torso Design

(a) Nyku V2 Body; Shown without horizontal ribs.

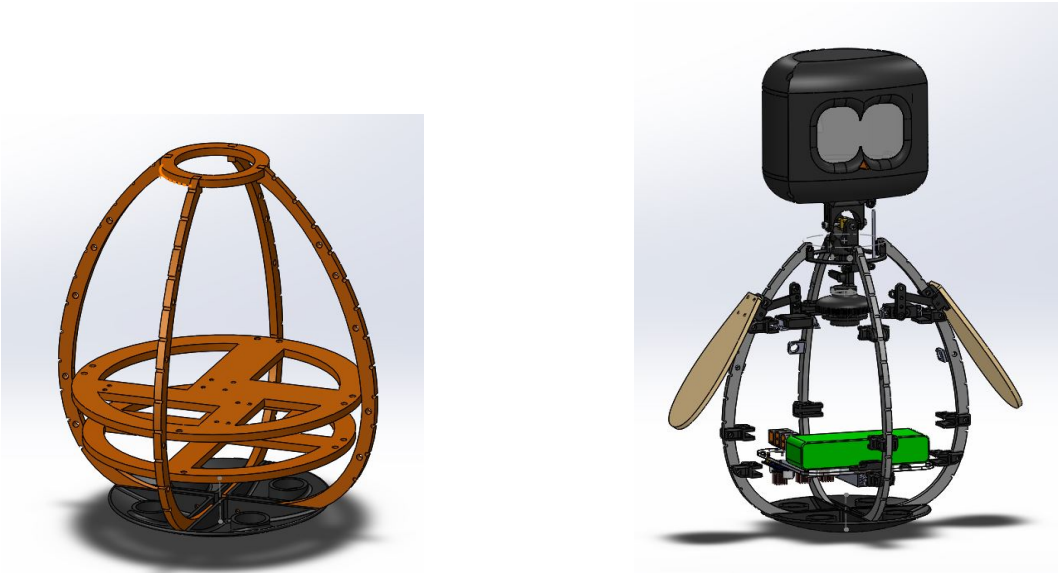
(b) Omni-Wheel Base.

Figure 3.3: Nyku System

First, we consider the torso, which houses all of the necessary components for Nyku's functionality and provides interfaces for the base and neck joints. In the torso of the robot, we house the main computer, a Raspberry Pi (RPi) 4B+ , as well as the peripheral controllers for the neck and the flippers. The torso also houses the power supply for the untethered system. Using a staged layer method we are able to separate power supply and electronics into 3 distinct layers in the torso. By isolating the control and power hardware, rapid iteration of the base structure was made possible. This allowed Nyku's form factor to be changed 3 times relatively quickly. Below, in Figure 3.4, we see the first two iterations of the body structure. V0 was a good starting point but was never used for concept testing as the layer system was not designed to be removed, and as necessary components began to materialize, the size became restrictive.

During the initial phase of design, the idea was to control Nyku's torso orientation internally, allowing the robot to be totally portable system. The concepts for these systems will be discussed below, and their problems and limitations explored.

In the first iteration of the V1 frame, the torso tilting was achieved by physically moving the center of mass (COM) of the robot internally using an inverted pendulum mechanism, visible in Figure 3.5. Here the purple piece



(a) Nyku Body V0. (b) Nyku Body V1.

Figure 3.4: Previous Nyku Body Iterations

in the center supports a carbon steel weight and is controlled by two servos, which individually push the weight in the horizontal XY plane to displace the COM of the robot. An additional view is provided in Figure 3.6, where joint J represents the spring, RMR represents the manipulated mass and $S0$ & $S1$ are the servo controlled joints. Initial testing found that the system worked best by imparting dynamic loads to the robot, meaning that position could not effectively be held. It also indicated that in order to affect the COM by changing the position of a ballast in the frame, the ballast itself had to be a majority of the weight of the frame. Taking this, and the additional weight of the other necessary components into account, the mechanism to move the

ballast around would have to be rigid enough to support the dynamic loads of ballast. With this concept, the design of the system would greatly increase in complexity and weight, counter to the goals of inexpensive manufacturing and portability.

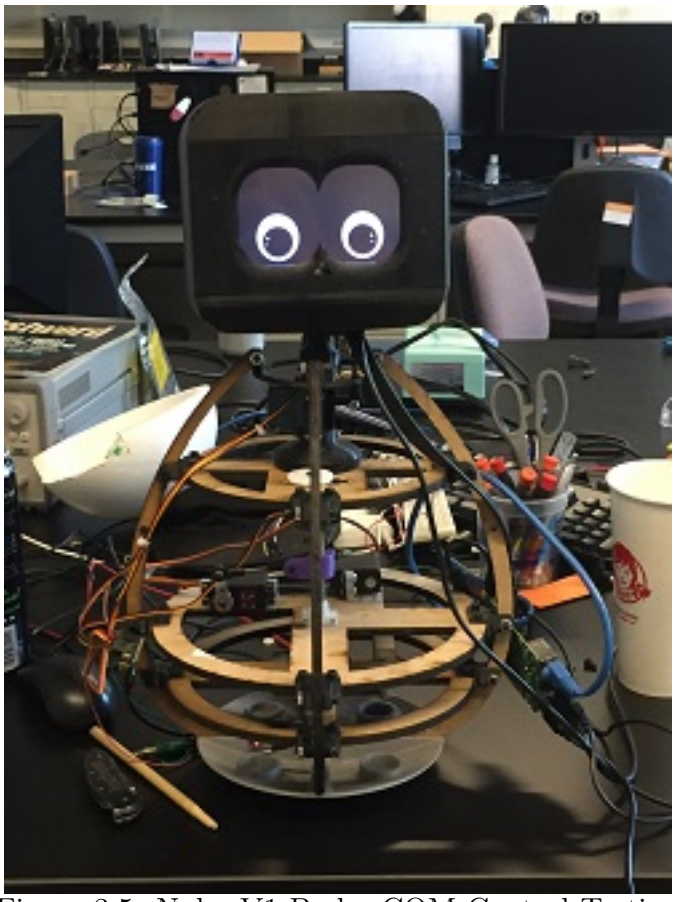


Figure 3.5: Nyku V1 Body; COM Control Testing

Once this concept was scrapped, the next iteration of internal posture control was to implement a control moment gyroscope (CMG) in Nyku's torso.

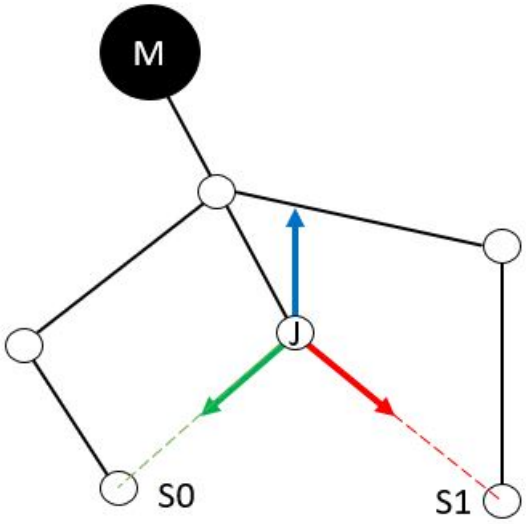


Figure 3.6: Center of Mass manipulator diagram

The idea here was to use the angular momentum generated by a spinning flywheel to adjust the orientation of the body. This system is commonly used to adjust the attitude of satellites in space [47]. The principle of the CMG system works on the extrapolation of Newton's second law into rotational motion. As seen in Figure 3.7, the internal flywheel imparts its angular momentum on the first gyro plate through the center of its rotation. This gyro plate, which is also revolving, has its own angular velocity which, when observed from the second gyro plate through the rotation about Axis 2, there is a combined multi directional angular velocity of both the flywheel and the first gyro plate rotation. This is finally attached to the second gyro layer, which again rotates with an

angular velocity. When this is all said and done, the Nyku Attachment layer sees a combined angular velocity from the motion of the flywheel and both gyro plates. When any of these bodies impart angular acceleration then, the Nyku attachment layer, and through it Nyku's body, experience a torque given in simplified form as $T = I\alpha$. Where T is the 3D torque vector, I is the moment of inertia tensor for the system and α is the 3D angular acceleration vector of the system. While CMG systems are well studied and are very useful in space, where the external torques due to gravity can be omitted, using such a system to hold the orientation of Nyku becomes unrealistic. With this internal momentum control, yaw about Nyku's vertical access was achievable, but in order to hold a position torque would need to be constantly applied, meaning that the CMG motors would need to be constantly accelerated, rendering the concept impossible to achieve as the motors would eventually saturate and reach full speed. Additionally, in initial testing, a flywheel broke off the shaft spindle. The flywheel then broke into pieces sending the bolts on the perimeter flying, puncturing a students monitor. This presents a serious safety issue for use with children. In order to use a CMG, a safety mechanism would have to be implemented, again increasing the complexity of the design. Because of these reasons this concept was also abandoned.

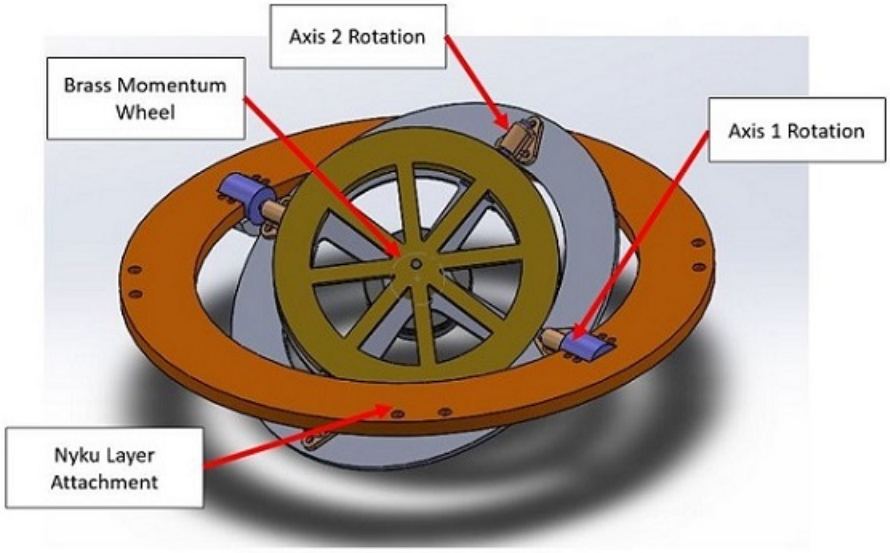
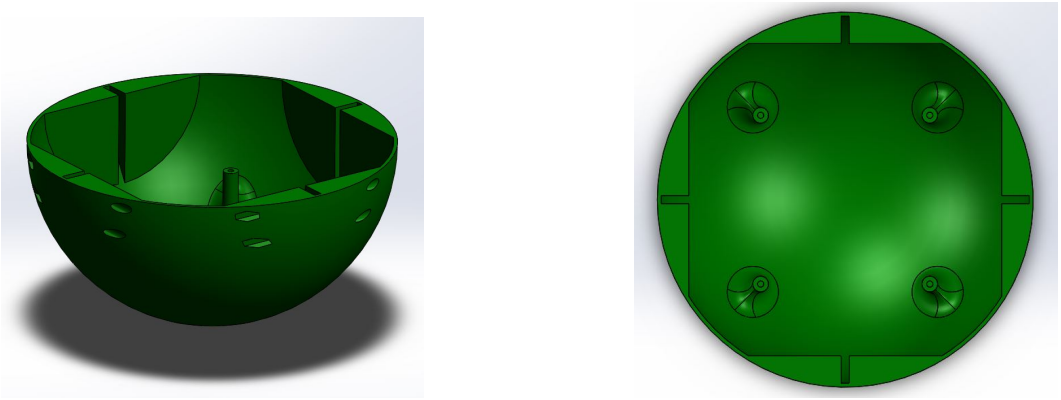


Figure 3.7: Gyro Plate Layer Concept for Internal Momentum Control

After this concept was exhausted, the idea to control Nyku's torso orientation using internal mechanisms was discarded and external position control was adopted. Taking inspiration from Star War's BB-8, controlling the spherical base with omni-wheels became the clear path forward. After this was decided, the structure of Nyku's torso was finalized into the most recent iteration of the body, as seen in 3.3a. Although it follows the same design principles using a round base, 4 vertical ribs, with horizontal layers housing components, V2 became a larger and more fully fledged design as the control system was decided upon. In V2, the round base becomes the full

hemisphere instead of just a cap and the balancing of the components is taken into consideration when mounting to the hardware layers.

The 12 inch diameter hemispherical bottom of the body is the interface between the Omni-wheel base, as well as the structural element supporting the vertical ribs. On the sides of the hemisphere there are holes that form a chord passing through the interface slits for the vertical ribs, visible in 3.8a. These holes are tapered to aid in the insertion of hardware. One side of the hole is a hex fitted to a 6-32 thread hex nut for each of assembly, while the other side is circular, accommodating the 1.5" 6-32 screw itself. From the top, Fig. 3.11b, we can see the 4 vertical slits that house the ribs. The ribs themselves provide mounting points for the hardware layers and at the top they are connected by the neck mechanism, which will be covered in the following section.



(a) Angled View (b) Top View

Figure 3.8: 12 Diameter Hemisphere base (Colored green for clarity)

The ribs are both a structural and aesthetic element of the design. As mentioned, they are used for structural support of the internal hardware, but they also lend to the oval shape of the robot, making Nyku look like a penguin. There are two configurations of the vertical ribs, both with and without the mounting holes for the servos that actuate the flippers, Figure 3.9. The vertical ribs also provide an interface for the horizontal ribs of the robot which define the round shape of the body. These horizontal ribs are made of 0.25" PTFE tubing. Using this material allows Nyku to have a pliable outer texture to facilitate a nice tactile sensation desirable in toys.

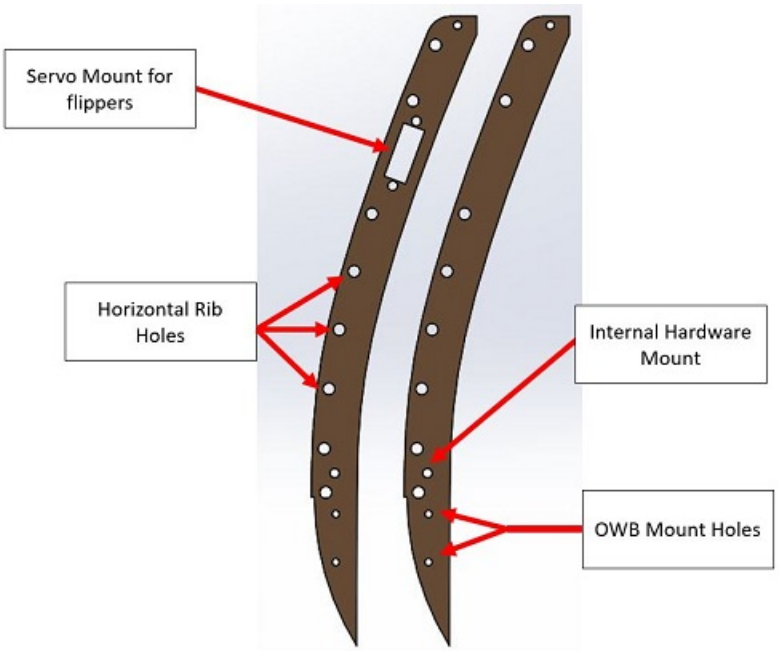


Figure 3.9: Vertical ribs, both configurations shown

The hardware layer contains the power supply for the untethered system and the control hardware for the neck and head. This layer contains two stages, on the top stage there is a powered USB 3 hub which provides power and data communication for Nyku's peripherals such as the microphone, speakers, screen for the eyes, communication hub for the dynamixel servos. This is also where Nyku's speakers are mounted. The bottom stage houses the battery, the Raspberry Pi and Maestro Servo controller for the flaps as well as the power supply circuit, which will be discussed in the electrical design section.

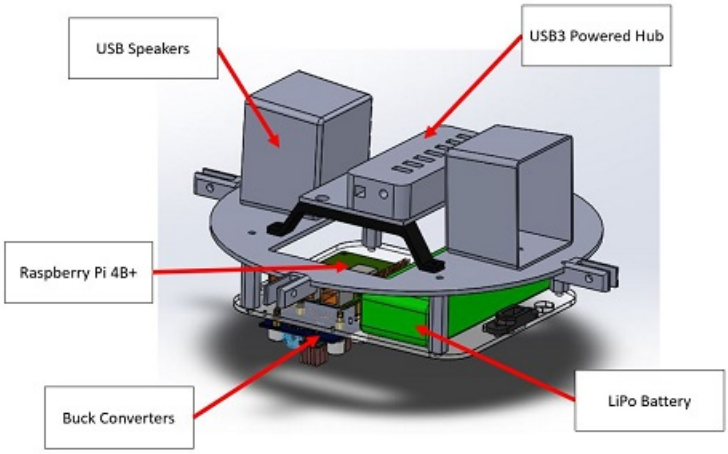


Figure 3.10: CAD Model of Hardware Layer

Originally, the torso was meant to be covered by a crochet sweater, however due to the unusual pattern it was difficult to plan construction for this method of manufacture. Instead a 4 panel sewed covering was designed. Using Solid Works surface projection, the quarter panels of the sewed covering

were designed. This was done by revolving the outer perimeter of the vertical ribs 45 degrees to represent the desired shape of the body, see 3.11a. This formed the quarter panel, next a plane tangent to the quarter panel shape was created and the outer perimeter of the quartered body was projected onto this plane to form the sewing pattern. A seem allowance of 1cm was added to the outer perimeter of this shape. This pattern was cut out of cardboard as a template and used to transfer the pattern to fabric. A few iterations were made using black scrap fabric, an early iteration can be seen in Figure 3.1, however the final version was made out of stretch velour, which is a velvet knit with polyester and spandex. This allowed the form to fit the shape better and allowed for more errors in sewing. For future development, someone with more sewing experience should be consulted in order to produce a repeatable pattern and suggest how to thicken the fabric to hide the horizontal rib lines.

3.2.2 Neck & Head

Moving up the body we arrive at the neck mechanism. This mechanism serves as the joint between the torso link and the head link, L1 and L2, and as such must meet certain requirements. Mainly being that it must withstand the loads of the heads motion and do so in a way that facilitates smooth motion.

(a) Sewing Pattern Extraction from Rib Shape

(b) Cardboard Template for Fabric Transfer

Figure 3.11: Nyku Outer Covering Sewing Patterns

In engineering terms, the structure must be rigid to the dynamic loads of motion and the controlling motors must not approach their maximum dynamic torque when accelerating the head. As such, the design of the neck was also an iterative process, whose requirements were validated experimentally. All iterations of the neck share common design elements. Each version has a universal joint allowing for pitch and roll, and they sit on top of the yaw axis. In the first iteration, control motors were common hobby servos in an attempt to reduce cost and allow for easy control. The first version used a

ball & socket spherical joint to allow for pitch and roll, however, this did not constrain the motion well enough and the application of pitch and roll from the servos resulted in unwanted yaw in the head. So this ball & socket joint was changed to a more common universal joint. Another issue was that despite the torque ratings of the servos, they struggled to move the head smoothly so they had to be changed as well. The head, mainly designed for aesthetics, has also gone through an iterative design process, but its changes have been minor and only made to accommodate new hardware and for aesthetic purposes. Mainly, the bottom cap has been altered to account for the flexible neck rather than hooking into the knit sweater concept.

Below in Figure 3.12, the final iteration of the neck and head design is shown. This is a modular design which bolts to the top of the ribs and holds them together solidifying Nyku's structure.

The neck mechanism is designed this way in the event that the body shape needs changing. It is a fully contained system which houses 3 Dynamixel AX-12A servos for decoupled control of the roll, pitch, and yaw of the neck. The neck consists of a bottom plate, which offsets the yaw servo from the rotating shaft of the neck, via the gear train visible in Figure 3.13. This is done to isolate the yaw servo from the axial load of the necks weight, and the radial load of the necks motion. Instead, the vertical shaft of the neck

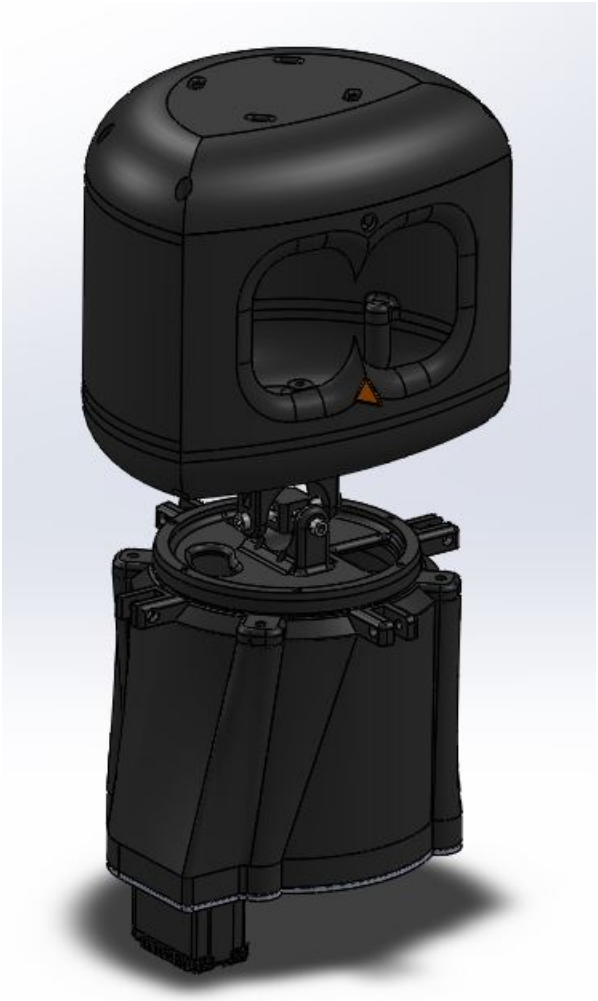


Figure 3.12: CAD Model of Neck & Head Module

is constrained in the bottom plate by two bearings of different sizes. The vertical shaft itself is assembled from the two dynamixel servos end to end. This allows the axis for roll, pitch, and yaw to intersect which aids in realistic motion of the head as well as simplifying the orientation control of the head, see Figure 3.13.

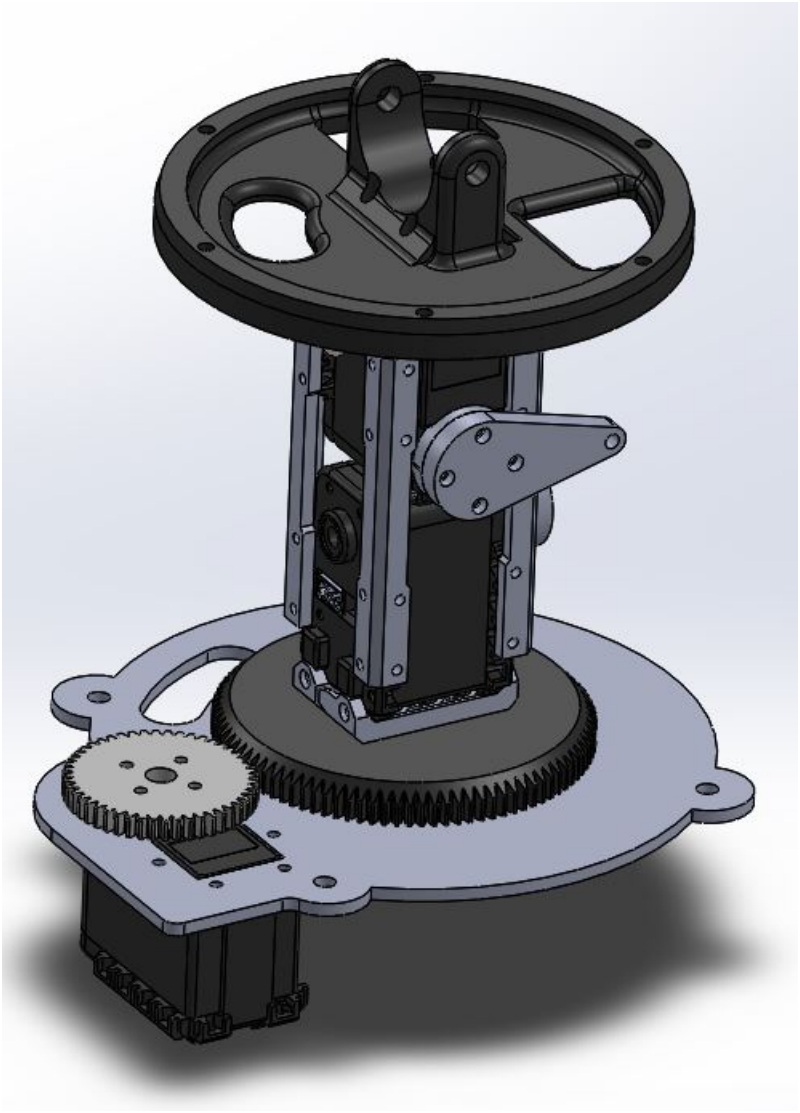


Figure 3.13: Neck Control Shaft

In the SolidWorks model, the links connecting the head are omitted as the spherical joints on the ends of the links cause rebuild errors. In practice, the links are made from 6-32 threaded shaft connected to spherical joint heads that allow the length to be tweeked by unthreading and threading on the

shaft. Links are therefore measured from the SW model and tuned by hand. As seen in Figure 3.13, the neck shaft is a modular unit containing both bearing interfaces and creating a module that can be disassembled without having to remove the control rods. This module is housed within a cowl that forms the outer shell of the Neck & Head module. This housing serves as an interface to tension the bearings that constrain the bottom of the Neck Control Shaft, this is demonstrated in Figure 3.14. The housing also serves to quiet the pitch and roll servos during operation for a cleaner user experience.

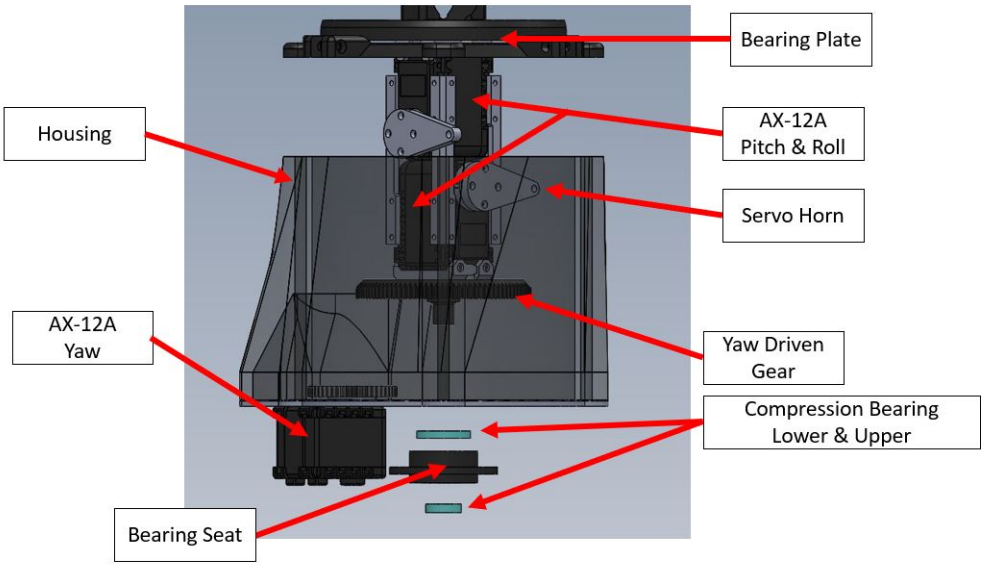


Figure 3.14: Exploded view of Neck Control Shaft

To constrain the top of the Neck Control Shaft, there is a bearing plate system, also highlighted in Figure 3.14. This consists of an upper and

lower race filled with loose 5mm steel ball bearings which are compressed by tightening the lower compression bearings. This mechanism is similar to how the headset on a bicycle works. The upper race of the bearing plate has cutouts to allow for the roll and pitch control rods, as well as the wiring for components mounted in the head.

The head itself sits on a 3D printed U-joint atop the bearing plate. This 3D printed u-joint has bearings pressed into the 4 side walls of the cube, which are compressed together by M4 dog-nose screws which when tightened into the heated thread inserts form the axles of the u-joint. This can be seen in Figure 3.15. The head itself provides an aesthetic housing for the screen used to display the eyes, the microphone to listen to the subject, and the LED for the nose.

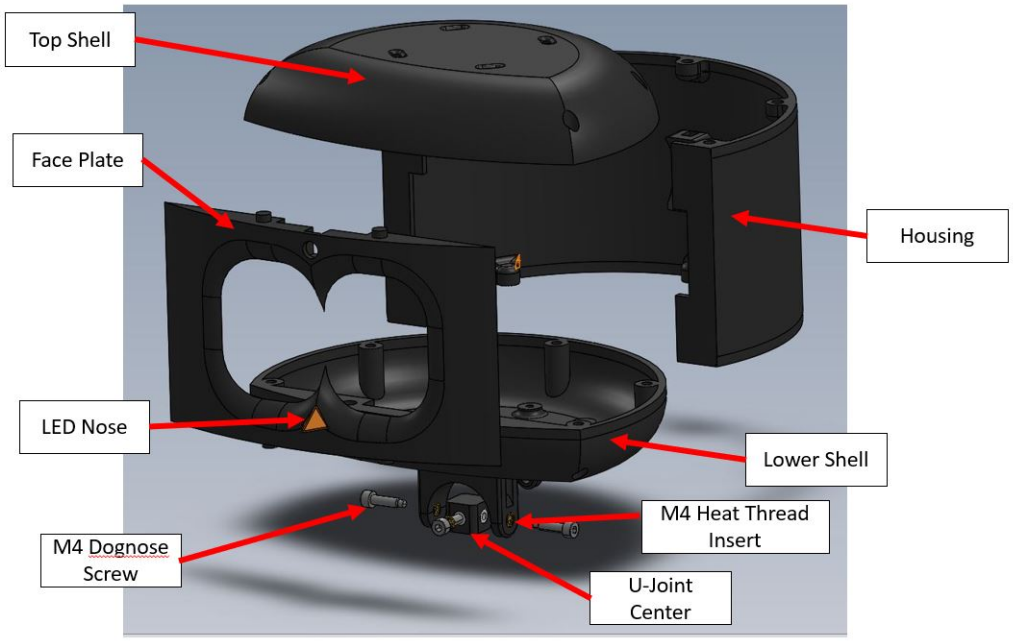


Figure 3.15: Exploded view of the Head Assembly

In order to achieve the aesthetic goals of the design, the neck mechanism was covered in a flexible rubber bellows printed in TPU. This part is omitted from the SolidWorks assemblies as the flexible nature of the material cannot be modeled for the motion. However it is presented in Figure 3.16. The top neck bellow, in 3.16a, bolts to the bottom of the head, and the bottom neck bellow, in 3.16b, bolts to the top of the bearing plate. Together, these flexible bellows overlap and cover the control rods of the neck, the u-joint mechanism, and the wiring for the head components, thus preserving the penguin aesthetic.

(a) Top Neck Bellows (b) Bottom Neck Bellows

Figure 3.16: Nyku Outer Covering Sewing Patterns

3.2.3 Omni-wheel Base

The Omni-wheel base is the interface between Nyku's torso, L1, and the inertial link of the system, Figure 3.2. This spherical joint is created by interfacing a hemisphere with 3 omni-wheels sitting tangentially on its surface. Using this configuration, the joint is able to perform infinite rotations about any axis passing through the center of the sphere. In the case of Nyku, these rotations are only limited by the hemisphere that makes up bottom of the torso.

Initially, the component was designed with 4 omni-wheels which balanced the sphere between them, as seen in Figure 3.17. This system relied on high power servos to control the orientation. These servos are implemented by DC motors geared 1:1 to the parallel shaft of a 5 turn rotary encoder. These motors are elevated at a 45° and are spaced out by 90°. In this configura-

tion, pitch and roll are actuated by 2 motors in the same plane in opposite directions, which simplifies the control of the system greatly. Because the system has three rotational degrees of freedom and four motors to control them, the system is over actuated. Yaw is controlled by all motors moving int he same direction. The rotations on the encoder enabled the full range of roll and pitch needed for Nyku's orientation, but had to be zeroed before the Torso was placed on the base, which is not a preferred design as it complicates startup procedure for the robot. Despite not limiting the motion in roll and pitch, using the analog potetionmeters for feedback did limit the possible motion in yaw. In addition to this, there was a flaw in the design, as the geared shafts were cantilevered, allowing for deflection at the of the potentiometer's travel where the gear would slip and the system would lose its position.

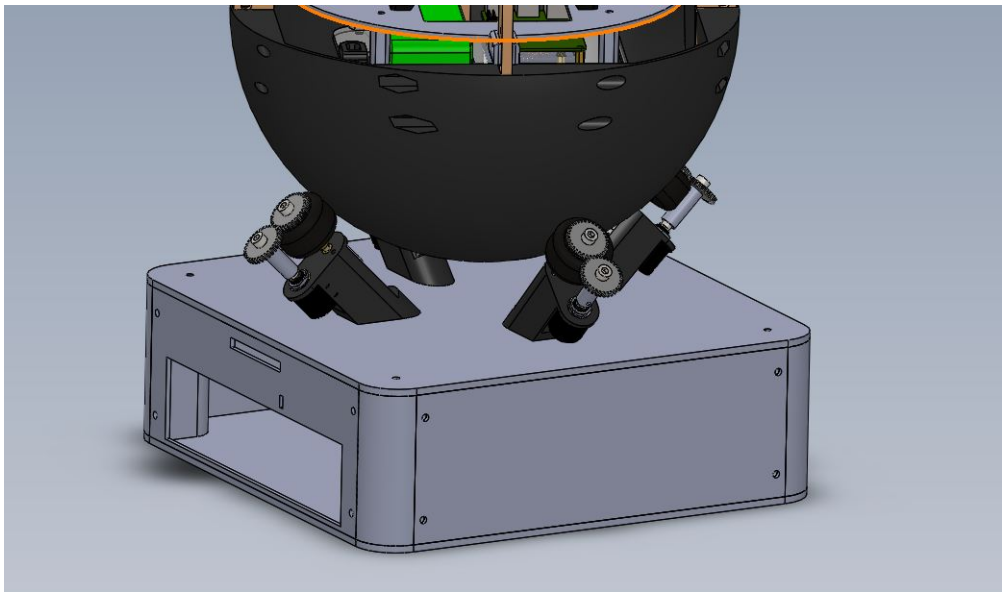


Figure 3.17: Omni-wheel Base V1

As seen above, the servos sit above the base, which is hollow to allow for the motor control unit and power supply for the system, however it leaves the mechanics exposed. This presents safety issues when interacting with children. Due to these errors, the base was redesigned and implemented as in 3.3b. In this design, the motors are still positioned at an elevation of 45°, however the radial spacing is 120°, allowing for three servo systems instead of four. In this case, the system is properly actuated, but position control becomes a much more complicated problem, these details will be discussed in Chapter 4. In Figure 3.18, the hemisphere is elevated from Omni-wheel Base to illustrate how the interface works. The three servos support the hemisphere at the 45°

tangency point on the sphere. When the robot is placed on this interface, it flexes the acrylic bottom layer and can decrease the elevation angle of the omni-wheel servos, to alleviate this an aluminum reinforcement is bolted in between the acrylic bottom plate and the servo assemblies. This helps to preserve the kinematic model of the system used in control.

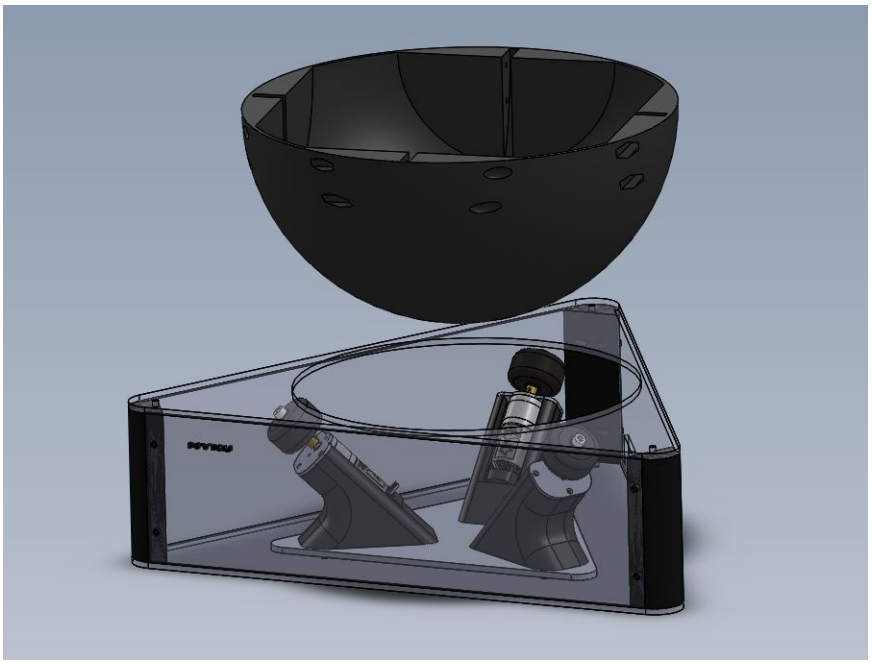


Figure 3.18: Omni-wheel Base V2 with the Hemisphere Elevated

In the second version of the Omni-wheel Base, the omni-wheel servos, in Figure 3.20, are constructed using encoded Pololu Gear Motor #4867. These are constrained using the 3D printed mounts, which set the elevation angle and bolt into the acrylic base to form the radial spacing. Using encoders

for the feedback allows for a cleaner omni-wheel servo overall, and allows for automated zeroing of the body, because the IMU can be used for orientation measurement and the encoders can be reset correspondingly.

Figure 3.19: Omni-wheel Servo V2

Mounted to the Pololu motor, are the omni wheels themselves, pictured in Figure 3.20, are mounted to the D-shaft of the motor via set screw. These wheels have a outer diameter of 38mm. Given the diameter of the Hemisphere, in Figure 3.8, is 12 inches, the gear ratio between the omni-wheel servo an the base is approximately 8. Using this gear ratio, the rotations of the omni-wheel servo can be mapped to an axis that is coincident with the center of the hemisphere.

Figure 3.20: 38mm Shaft Mount Omni-wheel

Version 2 of the Base, contains the omni-wheel servos inside the triangular acrylic housing. This prevent any pinch zones that may have been present in the first version and creates a cleaner look for the final product. However, a flaw of this system is that the motor control unit must be housed outside of the sleek acrylic case. This housing has yet to be designed.

The Body, Neck & Head, and Omni-wheel Base assemblies outline the major components in the Nyku system. They have been outlined in detail here so that the design can be understood, however, complete manufacture of Nyku will require study of the SolidWorks designs included with this document.

3.3 Electrical Design Overview

In order to make use of the hardware included in Nyku's design, the system must be wired for control and power. The decoupled system architecture mandates that Nyku and its Omni-Wheel base require different power

supplies and control circuitry. First, the power supply of the torso will be outlined, followed by the circuits connecting the subsystems within the torso. Next, the motor control unit and its power supply will be outlined for the Omni-wheel base.

3.3.1 Torso Electronics

For the Torso to be a stand alone unit, disconnected entirely from the Omni-wheel Base, the system must be battery powered. To achieve this, a Li-Po battery is used to power the system. Due to the relatively long run times required, an hour, and the power draw of the system, a 4 Cell 14.8V 6.6Ah battery was selected. From here, the power supply circuit is split into parallel loops for the necessary component voltages, as seen in Figure 3.21. The components branch off from the battery passing through a 10Amp power switch that serves as the main power switch for the torso system. Some components are USB powered, such as the screen for the head and the speakers. These systems are connected to the powered USB hub running off of the 9V buck converter, which has maximum current higher current rating. This 9V circuit also powers the Dynamixel servos, which are the main power draw of the torso system outside of the RPi computer.

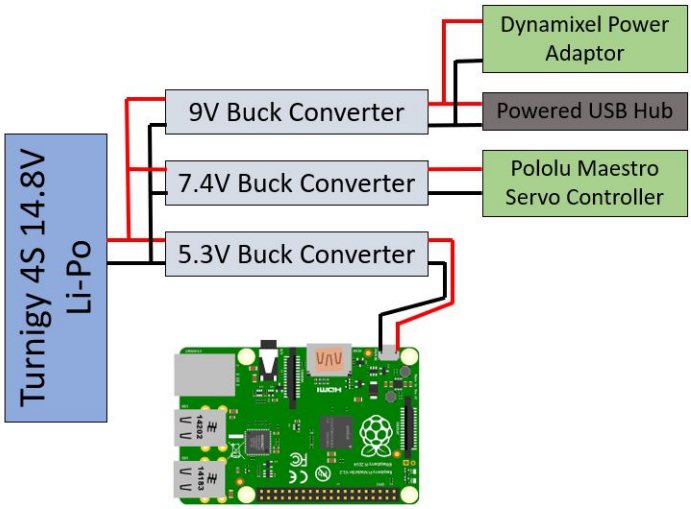


Figure 3.21: Torso Power Supply Diagram

Most systems in the torso are connected via USB which also provides power. Systems like the microphone and the Movidius Neural compute stick that perform computation are connected directly to the RPi via the USB 3.0 ports to eliminate possible communication bottlenecks.

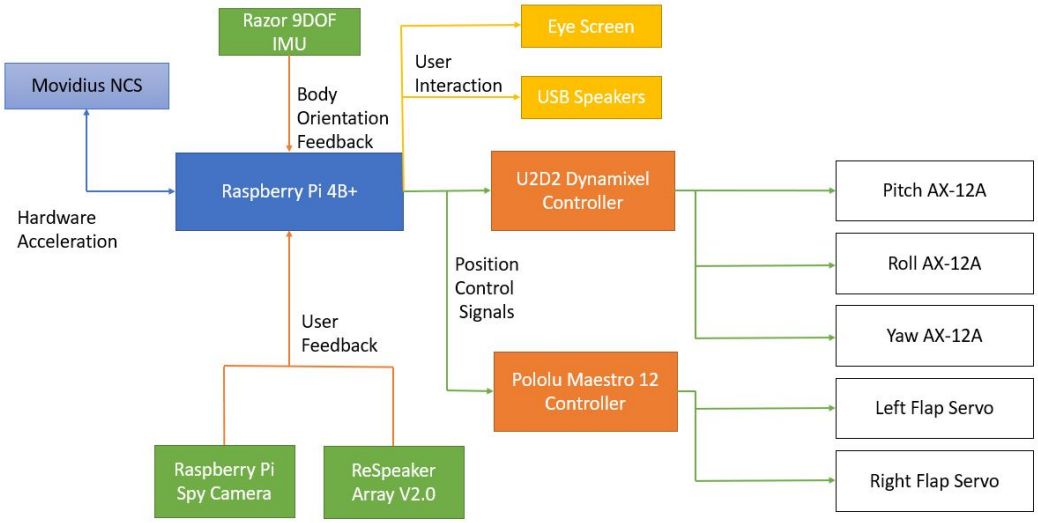


Figure 3.22: Torso Control Wiring Diagram

In Figure 3.22, the data connections to the torso hardware are diagrammed. Here, the RPi 4B+ is the central computer running the master ROS node. This connects to the IMU and U2D2 controller via USB2.0 and interfaces through a ROS Serial protocol. Currently, the IMU remains unmounted to the torso system as it is being used for the testing described in Chapter 4. The U2D2 controller then interfaces with the AX-12A Dynamixel Servos and the power adaptor through its proprietary 3 pin connectors. The Pololu Maestro is also connected via USB through the hub and control is done through a Python interface. The USB speakers and Eye screen are powered through the USB hub but communicated with through standard 3.5mm Jack and HDMI respectively. The spy camera connects directly to the DPI camera

port on the Raspberry Pi, while the ReSpeaker Array connected via USB3.0 directly to the RPi. Because the ReSpeaker is a micro-controller that performs direction of arrival and speech detection calculations on board, this system must have rapid communication to the RPi. The Maestro controller, provides a 3 pin DuPont interface standard to hobby servo connections, where the left and right flap servos are connected. These servos, HiTech HS-5087MH, have an upper limit of 7.4V and are powered directly from the Maestro Controller which is therefore is powered from the 7.4V source as seen in Figure 3.21. This details the wiring present in the torso which allows for the control of the individual system parts. Because this system is untethered, the power draw is tested under certain operating conditions so that total run-time can be more accurately predicted. Testing is conducted by measuring the max current drawn from a power supply under certain operating conditions and the amp hour rating of the battery is used to estimate the total run-time. For conditions including stall currents for motors, the stall current is added to the measured idle for the calculation. This data is presented in Table 3.1

Table 3.1: Run-times Calculated based on 6.6Ah battery

Condition	**Amps**	**Run-time (Min)**
System Idle	0.56	707.14
Movidius Obj D	0.79	501.2
Eye display and Object Detection	0.82	483.9
Speech, Obj D, Eyes	0.85	465.8
Speech, Obj D, Eyes, Motors IDLE	1.33	297.7
Speech, Obj D, Eyes, Motors STALL	7.05	56.1

As seen in Table 3.1, even at the worst case scenario where all motors, Dynamixels and servos, are stalled, the system is able to run for 56 minutes. Because this stall condition is very unlikely to occur, it is safe to assume that robot will be able to run for at least an hour long therapy session.

3.3.2 Omni-wheel Base Electronics

As discussed, the Torso is an untethered system, while the Omni-wheel Base is not. As such, the base has a traditional power supply, converting outlet AC to 12V DC at a maximum of 15A. This power supply is indeed very simple as it only provides direct 12V power to the motor driver and the RPi and as such only uses one buck converter to create a 5V rail powering the RPi.

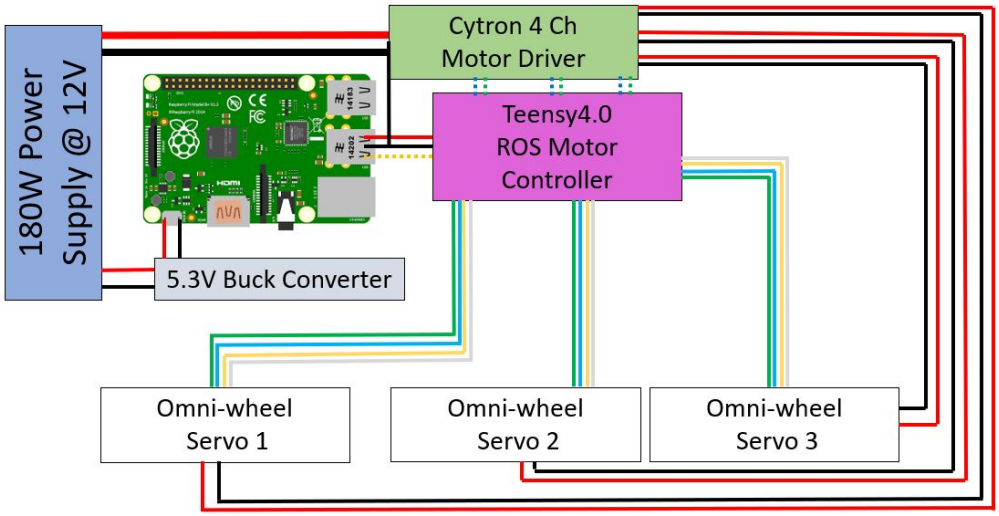


Figure 3.23: Omni-wheel Base Wiring Diagram

In Figure 3.23, the entire system is presented. Again, a RPi is used as the central computer, this allows for wireless communication over ROS protocol so that commands can be sent from the main RPi controller in the Torso. Over USB, the RPi powers the Teensy4.0 Motor controller and sends command signals through ROS Serial protocol. From here the RPi is not connected to any other systems and the Teensy4.0 interfaces with the Cytron FD04A motor driver. This is done via 20KHz PWM to control speed and direction is done via standard IO pin. Using 20KHz ultrasonic PWM allows for silent motor operation, although it is slightly out of spec for the motor driver. The PWM and direction pin are represented by the green and blue dashed

lines connecting the Teensy4.0 and the Cytron. It should also be noted that the Teensy4.0 and the Motor Driver share a common ground so that the PWM and Dir signals share a reference. The output of the motor driver is then the power to the motors, which can be seen exiting the Cytron block in standard red and black DC power lines which connect to the Omni-wheel servos. The shaft mounted encoders must then be connected back to the Teensy4.0 for closed loop control. Here the blue and green lines represent 3.3V and ground to the encoder, while the white and yellow lines are the encoders A and B channels. Using this circuit, the Omni-wheel Base is completed; accepting commands from the main RPi in the torso, sending them to the Teensy4.0 which does PID control for motor position, creating the Omni-wheel Servos.

This electrical subsystem went through many different iterations due to the difficulties associated with counting 3 quadrature encoders. At max RPM of the motors the quadrature data is being sent at 60Hz, which itself is not an issue, however when there are three motors rotating an issue could occur. Quadrature encoder counting is usually done with interrupt service routines on microcontrollers, where if A goes high it triggers an interrupt and adds to the counter, and the same occurs for B. If this occurs from two motors at once counts can be missed and the position data rendered invalid. In order to alleviate this problem originally, dedicated counter circuits which measured the

rotation of the shaft were used and read over SPI with an Arduino. However, these counter IC's proved to be unreliable in the configuration used and would only function for a few rounds of testing, so the design had to be changed. This is when the Teensy4.0 was selected as it has 4 dedicated quadrature channels with optimized interrupts that will not interfere with each other. The Teensy4.0 also boasts a processing speed of 600MHz, when compared to a standard Arduino rate of 16MHz the advantage becomes clear. However, the Teensy itself presents its own difficulties; the Teensy4.0 is a 3.3V logic system and the operating voltage of the encoders is from 3.5-20V with a preference for 5V. So initially the encoders were powered with 5V and level shifted down to 3V on a breadboard. This was done in two tests, one with standard voltage dividers and the others with a TXS0108E bidirectional level shift chip the initial testing, both of which worked flawlessly.

In order to make the design more compact and professional, this design was then migrated to a PCB where the TXS0108E chip handled the level shifting. This PCB design can be see in Figure 3.24. This PCB creates a clean interface between the encoders and the Teensy with integrated level shifting for optimal performance of the encoders. Drawing its power supply and reference from the Teensy via 0.1μf bypass capacitors, the TXS0108E chip shifts input level down to output reference value. In this case, it takes 5V logic from the

encoders and converts it to 3.3v logic for the Teensy. Using IO pin through a 33Ω pull down resistor, the chip can be switched into high impedance mode isolating the encoders from the Teensy during startup to avoid any unwanted signal spikes. This design was ordered from Oshpark PCB, and assembled into three test units.

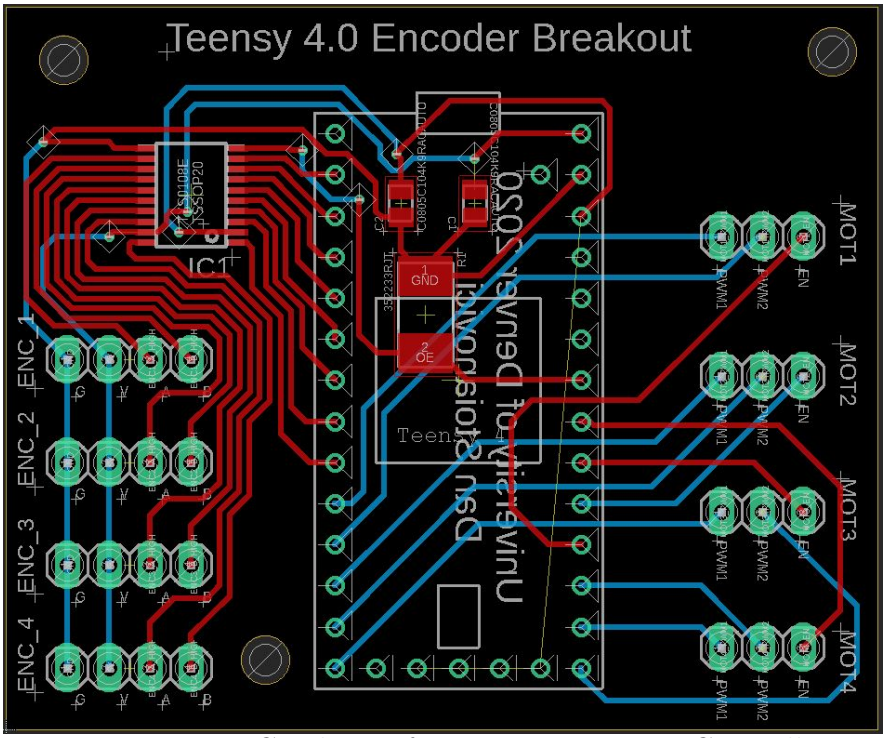
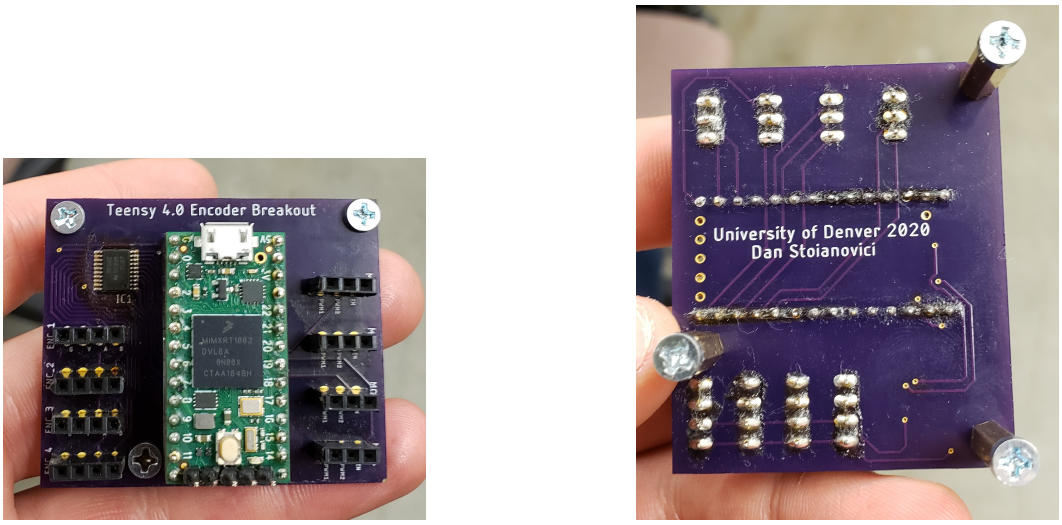


Figure 3.24: PCB design for Teensy4.0 Motor Controller

These boards were soldered with surface mount components in a reflow oven set to the correct temperature profiles. The final products can be seen in Figure 3.25. Then the headers are soldered on to interface with the motor

driver and the Teensy4.0. In testing, these units performed well for an average of three command cycles before the system would no longer work. This is despite following the requirements from the chips data sheet.



(a) Top of PCB w/ Teensy Mounted (b) Bottom of PCB

Figure 3.25: Encoder Breakout PCB

Due to time constraints, these issues were not solved, but the circuit was instead implemented on a perf-board, as seen in Figure 3.26. To avoid complex wiring, the encoders in this circuit are under powered with 3.3V instead of the datasheet minimum of 3.5V. This presented no issues during component testing and no counts were missed. In future development these issues should be remedied and the motor driver itself should be integrated into the chip making an entire control unit, this is discussed further in Chapter 5.

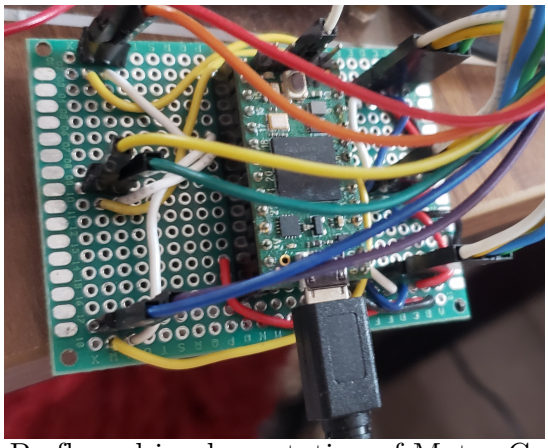


Figure 3.26: Perfboard implementation of Motor Control Circuit

3.4 Computer Configuration and Code

Now that the physical components of Nyku have been outlined and described, the computer architecture and the code to control Nyku is considered. First, the computer configuration that allows for the control will be explained. Second the control interfaces will be outlined and the flow charts presented for the control of the torso and neck. There is more code that runs Nyku, such as the Eye GUI, however this code is self explanatory. For any future work, all source code is provided in the GitHub, links to which are present in Appendix A. It should be noted that all code development was done using a Linux based laptop, as the compile times on the RPi are extremely slow, which would hinder the process. As such, when the final code is completed, the packages are to be cloned onto the RPi and only compiled once. Cross compiling for the

Raspberry Pi is a possibility, however it is not a well documented task and is outside of the scope of this thesis.

3.4.1 Computer Configuration

To code Nyku with the most modern robotics standards, the computer systems must be configured to accept ROS and OpenCV 4. To allow for optimization of neural network tasks and provide a platform for future development hardware acceleration must be made compatible with the system. To give Nyku the ability to play autonomously, a computer vision system must also be implemented, aided by the camera in the head and neural net based vision algorithms. In order to facilitate the use of these algorithms on the Raspberry Pi, the Intel Movidious Neural Compute Stick is used for hardware acceleration of predictive tasks, because it was readily available in the lab.

In the Nyku System, there are two main computers. One controls the torso, and the other controls the Omni-wheel base. In the Torso there is a Raspberry Pi 4B+ running a custom version of Raspbian Buster. Because the Raspberry Pi is an ARM based computer, there are not many prebuilt packages available to easily configure the system. So in order to run ROS Kinetic on the system, which at the time of development was the suggested LTS

distribution, ROS had to be compiled from scratch on the Raspberry Pi. This was a very involved process that required mismatching the compiler GCC and G++ versions so that some boost libraries could be compiled for arm. After this was done, OpenCV 4 had to be compiled from scratch, which presented a similar set of issues. Because ROS Kinetic does not support Python 3, only Python 2.7, OpenCV 4 had to be compiled following a very specific set of instructions. Currently, ROS Melodic is the suggested LTS distribution and base systems will have to be migrated in the future.

In order to use the Movidius NCS, the system had to be configured to run in an OpenVino environment, which allows for the forward processing of already made neural net models. This is achieved fairly easily using the documentation from Intel, and was not as daunting of a task as compiling the operating system. To test this functionality, a prebuilt MobileNet Single Shot Detector is run using the Movidius hardware acceleration, as seen in Figure 3.27. In these tests the RPi 4B+ with Movidius Acceleration is able to run the object detection model at 8.5 FPS.

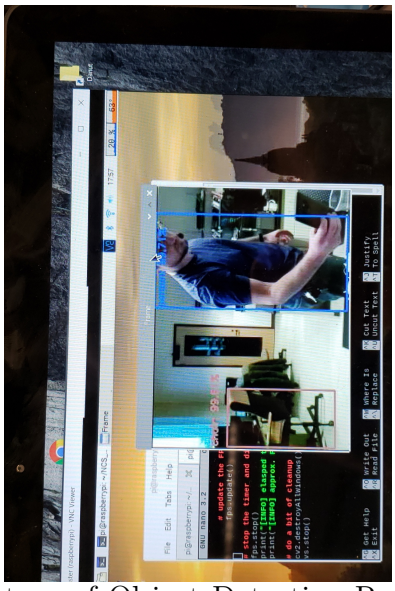


Figure 3.27: Screen Capture of Object Detection Running on RPi 4B+ with Movidius NCS

This operating system is then cloned and is used in the RPi 3B+ controlling the base. Now that both systems have ROS Kinetic, the ROS master URI of the Base's RPi 3B+ is set to the IP of the Torso's Raspberry Pi. In this way, the Base computer has access to all the ROS topics being published over the network and the systems are connected wirelessly. To make this process cleaner, there is a bash script that sets static IPs for both systems, and the Master URI, based on the name of the wireless network they are connected to. Once this is done the systems are now compatible with each other and are ready for the control pipeline.

3.5 Neck Control System

As the neck is built using Dynamixel servos, the control interface on the receiving side is already written. What is necessary is then to send the correct communications to the existing platform. To do this, the dynamixel_control_hw package is used to configure the hardware interface. Here a URDF file is used to name each servo, establish travel limits, define velocity limits, as well as establishing link connections, which must be stored in a URDF folder in the package. A YAML file doing the same is also required, but it is stored in the Neck control package (neck_dyna_test). In the neck_dyna_test, the control for the neck is done using all three methods exposed by the dynamixel_control_hw package. Included in the neck_dyna_test package there are examples covering control of the neck via an action client, a service client, and a direct communication with the command interface of the action client. The purpose of this package was to test which method of communication created the smoothest motion in the neck. It was found that interfacing with the action as a client provided the smoothest motion. This code can be found in src/neck_action_client_test.cpp.

As this is test code for the movement of the neck, this code does not take an input determining where the head should look. As such the code is

very simple, however it does expose methods to create the trajectory messages to send to the action server. First, the code establishes the ROS action client and its connection to the correct action server, in this case "/dynamixel_controllers/nyku_neck_controller/follow_joint_trajectory". From here, the program waits for the connection to the server, and then creates a goal trajectory from the previous location to the desired location. This method also takes the duration between points which allows for the specification of slow or quick movements. After this is done, the goal trajectory is sent. Before sending a new goal, the code waits for the result from the action server.

In order to run this example, one must first start the action server exposed by the dynamixel_control_hw package. This can be done by running the command "roslaunch dynamixel_control_hw nyku_neck.launch. Afterwards, one can simply "rosrun neck_dyna_test neck_action_client_test". This will execute a predetermined trajectory.

With these packages, and the example code, one has all the tools necessary to smoothly send positions to Nyku's neck, once the desired position is determined. After it was shown that the neck was capable of smooth motion, determined by observation, focus shifted to the control of the Omni-wheel Base, which proved to be a much more complex task.

3.6 Omni-wheel Base Control

In order to control the base, an interface for sending goal positions to the omni-wheel servos had to be determined. Unlike the Dynamixel neck, this interface was written from scratch.

To tackle this problem, first a PID controller had to be implemented on the Teensy4.0 which so that the position of the omni-wheel servos could reliably be specified. This is done using a standard PID algorithm, where the error between the desired and actual position is taken into account with the integral and the derivative of the error, each modified by their respective gains. In the PID controller, anti-windup for the integral term was implemented in order to keep the system stable during long periods of operation. A significant dead band tolerance was also implemented as the shaft encoders were more precise that allowed by the backlash in the gear train of the motors.

Initially, this PID controller was tested by manually sending goal positions via a serial monitor. As the Teensy4.0 can be coded in C++ with Arduino libraries, the serial communication protocol of Arduino was used to write a parser which could decompose a string of goal positions into an array of integers needed in the PID algorithm. This library is present in Teensy4MotorController package made available on GitHub (Appendix A).

However this method proved to have some issues. Although difficult at first, it was possible to send the correct string of goal positions from C++ using the termios library. This worked for initial testing, but when using the motor controller as an output to a main control pipeline, the motor controller on the Teensy4.0 had to report its status. When attempting to use the termios library to read from the port, timing the communication was extremely hard. The main pipeline would read the string of correct length, however it could not be synced to the start and end of the string being sent from the Teensy over serial. Without feedback from the motor position, the motion of the servos was discontinuous due to an arbitrary wait times included in order to ensure completion of a goal.

When researching to solve this issue, it was discovered that some microcontrollers could be coded to accept a rosserial interface. Although the Teensy4.0 was not part of the included controllers, very small changes to the rosserial_arduino package allowed the use the Teensy4.0 as ROS node, making it as robust as any other ROS enabled peripheral. With this the use of ROS topics on the microcontroller was possible. Thus the input and output pipeline of the motor controller on the Teensy4.0 was edited to receive an array of goal positions for the servos and return their error via an array of the same type. With this completed, control of the omni-wheel servos was finally reliable.

This source code can be found in the Teensy4MotorController repository in the ros-control branch. In order to run initialize the node, one must create a custom launch file specifying the serial port that the rosserial interface is looking for as well as the baud rate. Once the rosserial node is started it will automatically forward the topics to and from the Teensy4.0 motor controller.

Now that individual positions could be sent to the Teensy4MotorController, it was necessary to find what omni-wheel servo positions would result in the desired hemisphere orientation. This is discussed at length in Chapter 4, as the inverse kinematics are derived there. For the purposes of this chapter, it suffices to say that the joint angle calculations were implemented in a C++ library called eq_man_ik. Using this, the goal orientation of the hemisphere can be specified in RPY angles. To validate this a few test pipelines were established, and can be found in the nyku_omniwheel_base repository.

First, a Razor 9DOF IMU was used as an input to the pipeline to create the goal positions, which were then sent to the motor controller on the Teensy4.0. The node diagram for this pipeline can be see in Figure 3.28.

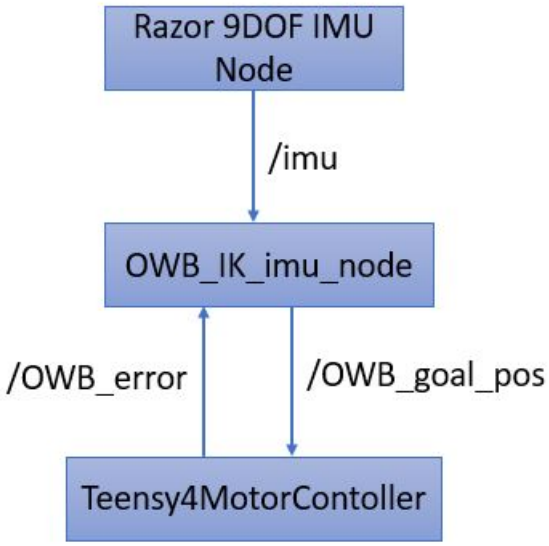


Figure 3.28: Node Diagram for IMU Control Pipeline

Here the OWB_IK_imu_node subscribes to the quaternion orientation from the /imu topic. This is the converted to RPY orientation in the callback for the razor IMU subscriber. This RPY orientation is then fed into the inverse kinematics calculator and the resulting joint angles are converted through the gear ratio of the omni-wheel to hemisphere interface, and then sent to the Teensy4MotorController which reports back the error. This flow is illustrated in Figure 3.29 While ROS is "OK" this loop continues and the hemisphere mimics the orientation of the IMU.

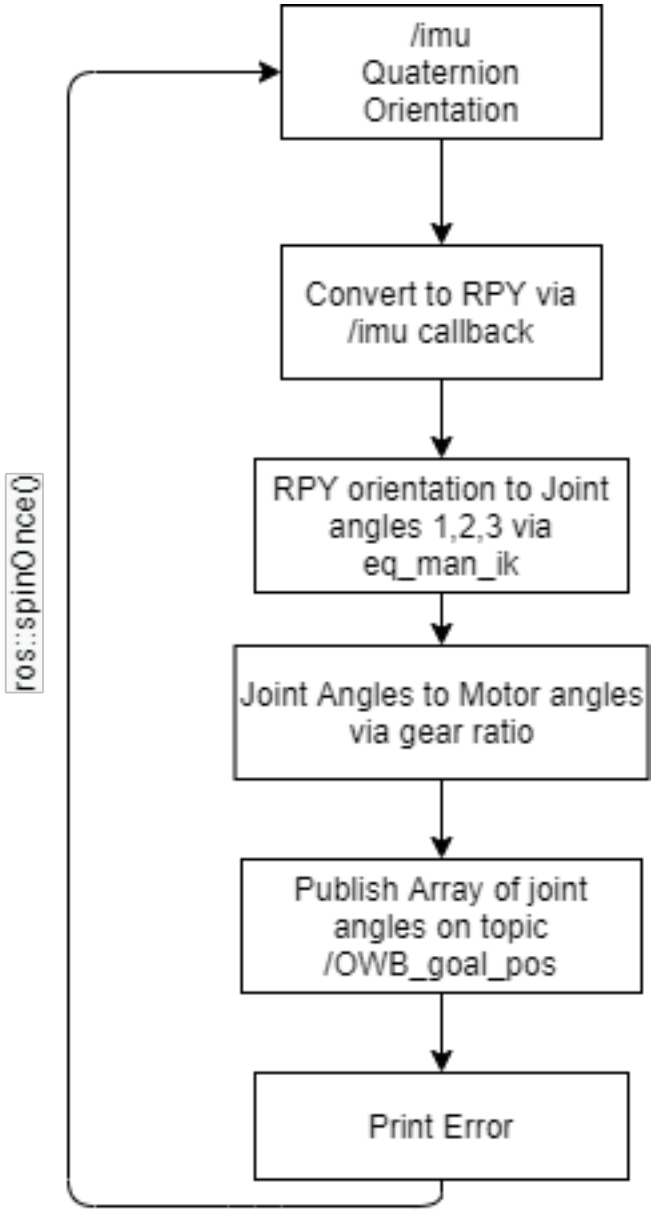


Figure 3.29: Pipeline Flow Chart for IMU Control

Similarly to the IMU controlled pipeline, the Skeleton tracking control pipeline takes quaternion orientation of the torso as seen in the node diagram,

Figure 3.30. Here, the OWB_IK_skel_node subscribes to the /torso_rotation topic and converts it into the necessary joint angles. Here however there is not a prebuilt ROS node for skeleton tracking. Skeleton tracking is implemented using the Nuitrack SDK and an Intel RealSense D435 RGBD Camera following their documentation. In /skeleton_track_node, the torso orientation is isolated and published to the /torso_rotation topic feeding into the pipeline in Figure 3.31

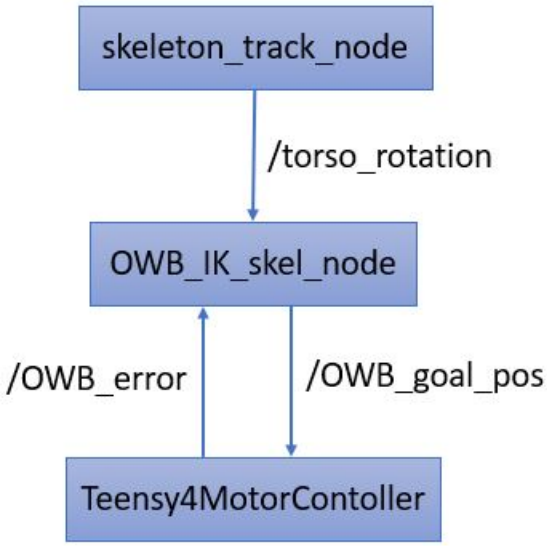


Figure 3.30: Node Diagram for Skeleton Tracking Control Pipeline

Although the pipelines for the IMU and skeleton tracking control look similar, there is a slight different in the quaternion conversions. The /imu

topic uses a sensor_msgs::Imu message as opposed to the /torso_rotation topic which publishes a geometry_msgs::Pose.

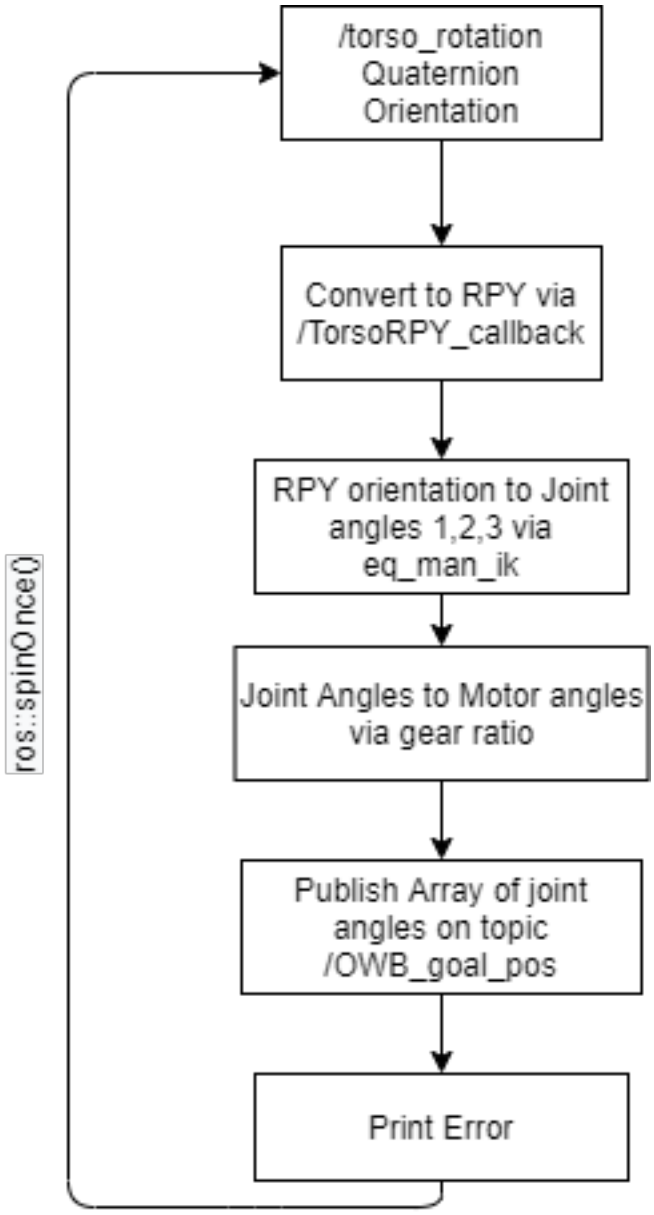


Figure 3.31: Pipeline Flow Chart for Skeleton Tracking Control

These were the preliminary tests run using the IK model described in Chapter 4. They were conducted in order to validate the control pipelines visually. Using the IMU to control the orientation of the sphere was very effective as there was a 1:1 mapping of orientation and it was intuitive. However, when the skeleton tracking was used to control the orientation of the hemisphere, the orientation of the Torso was difficult to map to the sphere. As the coordinate systems are not aligned, rotations about the axes were not mapped directly. This is to saw that torso yaw resulted in Omni-wheel Base pitch and so forth. This problem was exasperated by the lack of Nuitrack documentation describing the home orientation of the joints. Due to the three minute limit on the Nuitrack software and the difficulty recording a motion for playback this mapping was hard to determine experimentally.

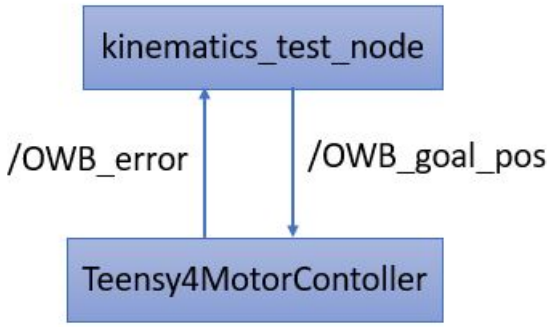


Figure 3.32: Node Diagram for IK Testing Pipeline

The first step in solving the above issue was to write a program to test the performance of the systems developed in this thesis and then try to solve compatibility issues with the Nuitrack software. As such, a test plan was developed using the following pipeline, as in Figure 3.32. The test procedure itself is fully describe in Chapter 4, however it also presented hear in Figure 3.33. In short, a CSV file of desired orientations is read into a vector of float vectors in C++ representing the desired RPY for a certain trajectory point. This goal orientation vector is then run through in order and its IK calculated, applied to the motors, and the resulting RPY is measured with the IMU. This IMU recorded data is then stored in a data CSV created each time the test program is run with a time stamped name for clarity.

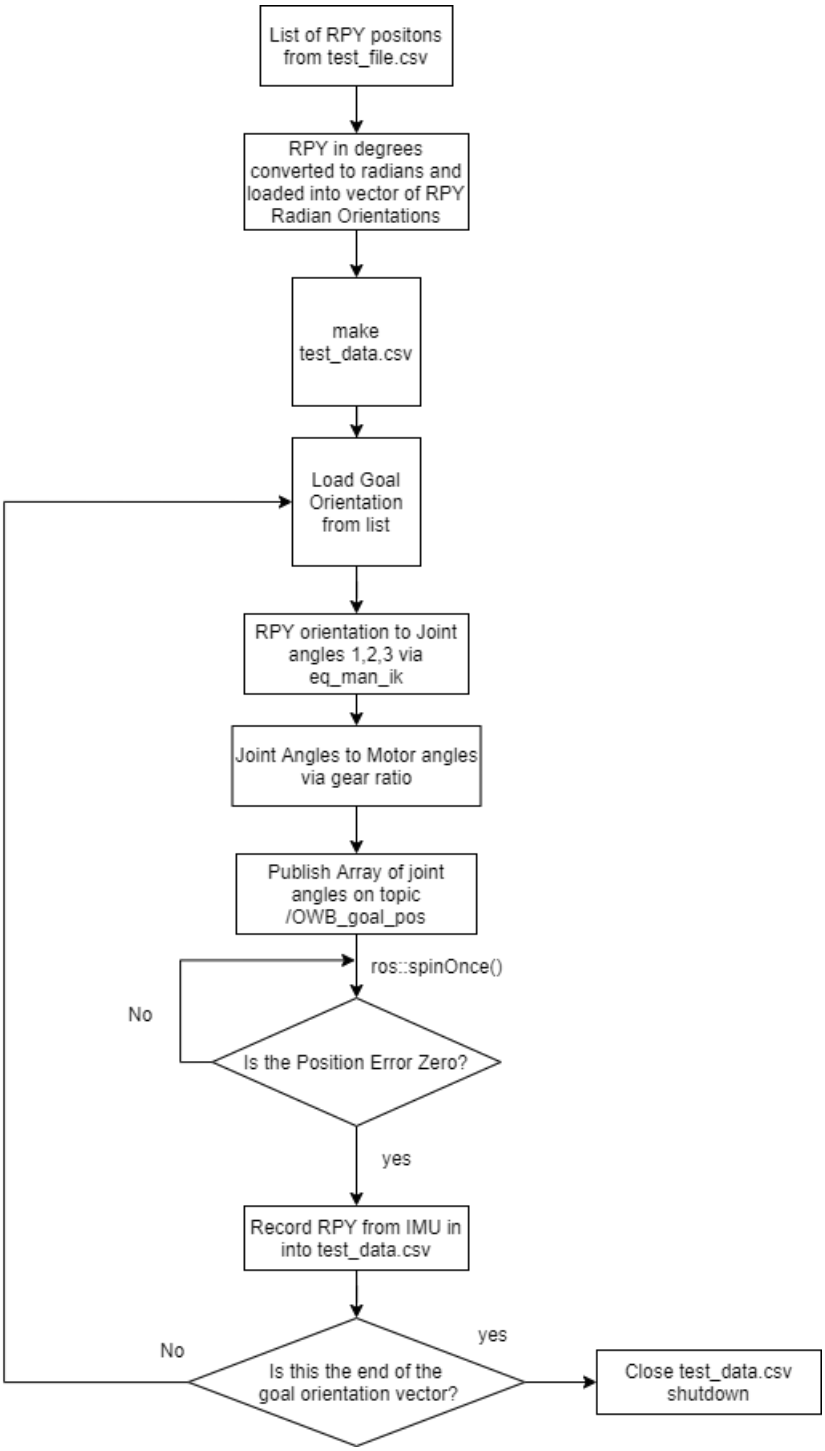


Figure 3.33: Pipeline Flow Chart for IK Testing

Chapter 4

Omni-Wheel Kinematics

4.1 Investigation of Omni-Wheel Base Control

The Omni-Wheel Base architecture for Nyku is modeled after many ball balancing robots that have been developed [32, 40], most influentially being, BB-8, the Star Wars phenomenon[50]. In these designs, the robot sits on top of a ball through a very similar Omni-wheel interface, which is an inverted version of Nyku's Omni-Wheel base discussed in the previous chapter. In these types of robots the controllable parameters of these systems are the angular velocities of the omni-wheels, not the positions of the omni-wheels. This is because the sphere is used for locomotion and the control is not concerned with the orientation of the ball itself. Notable works for this configuration are the ball bot developed by Dr. Danh Ngoc Nguyen and his team [40]. In their control scheme they are concerned with the orientation of the robot above the ball and the robots position in the navigable environment. They develop

the kinematics of the model using a Lagrangian method and achieve control using a Linear Quadratic Controller outputting wheel velocities. Similarly researchers at the Polytechnic School of the University of São Paulo have built their own ball balancing robot on a similar LQR method [32]. But again, these control schemes concern the position of the robot in the navigation plane, not the orientation of the sphere.

Another design, the Atlas sphere is developed more similarly to Nyku , where the ball sits on top of the omni-wheels [7, 28, 28]. This system developed by researchers at Carleton University was designed to be a 6 DOF simulator for testing equipment as well as training pilots. Their system mounts the sphere on an XYZ linear motion table as well, allowing for motion in all 3 Cartesian directions as well as rotations about all axes. Given the infinite rotation possible on the sphere, it is possible to simulate all forces that would act on the ball. The atlas sphere control scheme uses a mapping of tangential velocity from omni-wheel to sphere. This team has also extended their work to acceleration level kinematics, through simple derivation. However, when they tried to integrate the velocity level kinematics to position, the team was only able to achieve an estimate by numerically integrating quaternion rotations. This will be explored briefly in the next section as it is more applicable to the

architecture of Nyku's Omni-wheel base and inform the design of the proposed kinematic model.

In the development of the Omni-wheel sphere for Nyku, the design goal of posture mimicry dictates that position level control is better suited. And due to the computational restrictions of embedded micro-controllers such as the Raspberry Pi, an analytic solution is highly preferred. As such a kinematic equation is developed so that joint angles could be analytically solved given a goal orientation. To the best of our knowledge, this is the first analytical solution for the inverse kinematics of an omni-wheel/sphere mechanism's orientation.

4.2 Atlas Sphere Kinematics Background

In this section the kinematics proposed for the Atlas Sphere orientation manipulation mechanism and its test are explored to give background on existing work in the field of sphere manipulation. The kinematics for the atlas sphere developed by the researchers at Carleton University are meant to be applied on a sphere containing the test setup for pilots and satellites which requires the simulation of rotational forces. As such, the sphere is controlled in velocity and acceleration initially [44]. The velocity level kinematics are developed by mapping the tangential velocity of the omni wheel to the center

of the sphere by a transmission ratio of the diameter of both the wheel and the sphere. This solution can be derived into acceleration easily to control forces on the sphere. By using the simple velocity kinematics, the team was able to find the Jacobian of the system analytically, ensuring that the system is time invariant which serves to simplify greatly the acceleration calculations [7]. But as mentioned above, this method presents issues when integrating into position. In order to calculate position from the sphere, Beranek and Hayes integrate the quaternionic differential equation representing the velocity of the sphere. In practice this integration is done numerically and is therefor inherently computationally expensive. In addition to this, the solution is only valid given constant angular rates of the omni-wheels, which renders it practically invalid for position control given the need to start and stop motion once going to or arriving at a position. To test this method for the calculation of position kinematics, the team uses the algorithm to predict the orientation of the sphere based on input velocities to the omni-wheels and validates the orientation with a camera and markers that track the balls orientation. In their findings they do not report the error, however their figures indicate a strong lag between predicted orientation and actual orientation[7]. Due to the impracticality of this method and its dubious performance, a new kinematic model is needed.

4.3 Equivalent Manipulator Kinematics for Omni-Wheel Base

In this section the kinematics for the equivalent manipulator model of the Omni-wheel Base are discussed. This system is modeled as a parallel mechanism consisting of three serial link 3 DOF manipulators with intersecting axes, as seen in Figure 4.1. In this figure, we see all three manipulators contributing to the orientation of the last frame, represented by the checkered ball. Here, each manipulator is labeled; A, B, and C. This convention hold through this chapter. Each manipulator consists of three serial links, connected to a common base and common end effector. Here link 1 is green, link 2 is yellow, and link 3 is blue. As seen in this figure, the arms from different manipulators are intersecting. This illustrates the impracticality of actually building this system, and that it can only be used as a model for the system.

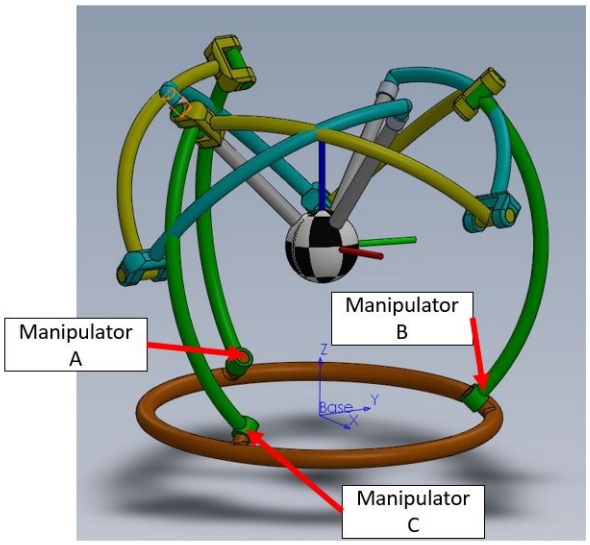


Figure 4.1: All three equivalent manipulators acting on the final frame

These manipulators are aligned according to the position of the motors in the base, 120 degrees apart at an elevation angle of 45 degrees, which are fixed angles defined by the structure of the base. From this orientation, the 3 DOF equivalent manipulator begins with its intersecting axes. The rotations caused by the manipulator are then projected back through the transformations of the motor structure and as such the rotations are referenced back to the home frame allowing for goal orientation to be defined in the intuitive inertial base frame. By modeling the Omni-wheel Base as a parallel mechanism of three serial link manipulators, each with 3 intersecting DOF, we are able to draw two conclusions. 1) That each manipulator of the parallel mechanism

contributes to the same goal, and 2) that because there are three consecutive intersecting axes, an analytic solution exists [13].

Armed with this information, we will now begin a discussion as to how the analytic solution of the 3 DOF intersecting axes manipulator is used to find the joint angles of the omni-wheel motors.

Equation 4.1 gives the forward kinematics for an individual manipulator in the home frame. As all of the axes of rotation intersect, as illustrated in Figure 4.2, all rotations can be expressed as 3×3 matrices, omitting the linear translations present in 4×4 rotation matrices. As such, R^{06} is a 3×3 generic rotation matrix for one motor mechanism contributing to the orientation of the hemisphere. The forward kinematics are assembled from the base to the end effector using the moving frame Euler X-Y-Z convention. Naming convention is as follows; R_z^{01} is the rotation from frame 0 to 1 about the Z axis.

$$R^{06} = (R_z^{01} R_y^{12})(R_x^{23} R_y^{34} R_z^{45})((R_y^{12})^T (R_z^{01})^T) \tag{4.1}$$

Where the individual contributing rotations are of standard forms as shown in Equation 4.2 with $s\theta = sin(\theta)$ and $c\theta = cos(\theta)$ of some arbitrary rotation angle θ.

$$R_x = \begin{bmatrix} 1 & 0 & 0 \\ 0 & c\theta & -s\theta \\ 0 & s\theta & c\theta \end{bmatrix} R_y = \begin{bmatrix} c\theta & 0 & s\theta \\ 0 & 1 & 0 \\ -s\theta & 0 & c\theta \end{bmatrix} R_z = \begin{bmatrix} c\theta & -s\theta & 0 \\ s\theta & c\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{4.2}$$

In Equation 4.1, the terms are split by parentheses into groups. In the first group ($R_z^{01}R_y^{12}$), we have the rotations concerning the orientation of the motor, where θ_0 is the rotation around the base Z axis corresponding to the 120 degree circular placement of the motor and θ_1 is 45 degree elevation angle of the motor. In the second group, ($R_x^{23}R_y^{34}R_z^{45}$), we have the rotations that represent the degrees of freedom the omni-wheel. Here, θ_2 is the motor's rotation angle about X_2. Although it appears that there is a linear displacement between X_1 and X_2 in Figure 4.2, these axes are actually coincident and the mapping between them manifests mathematically as the gear ration between the diameter of the ball and the diameter of the omni-wheel. θ_3 is the pitch allowed across the omni-wheel rollers about Y_3, and θ_4 is the less intuitive

yaw about Z_4 which represents the twist that can occur at the tangent point between the sphere and the omni wheel. All three of these rotations are made possible by the omni-wheel, however only θ_2 is controllable, while θ_3 & θ_4 are resultant based on the desired final orientation. In Figure 4.2, reference frames 2 through 4 are shown as a combined frame with axes X_{2-4}, Y_{2-4} & Z_{2-4}. The last group, $((R_y^{12})^T(R_z^{01})^T)$ is inverse rotation in order to return the final frame back into base coordinates so that it may be compared with the other 2 manipulators allowing for the specification of only one goal position in the control system. These rotations are clearly illustrated in Figure 4.2.

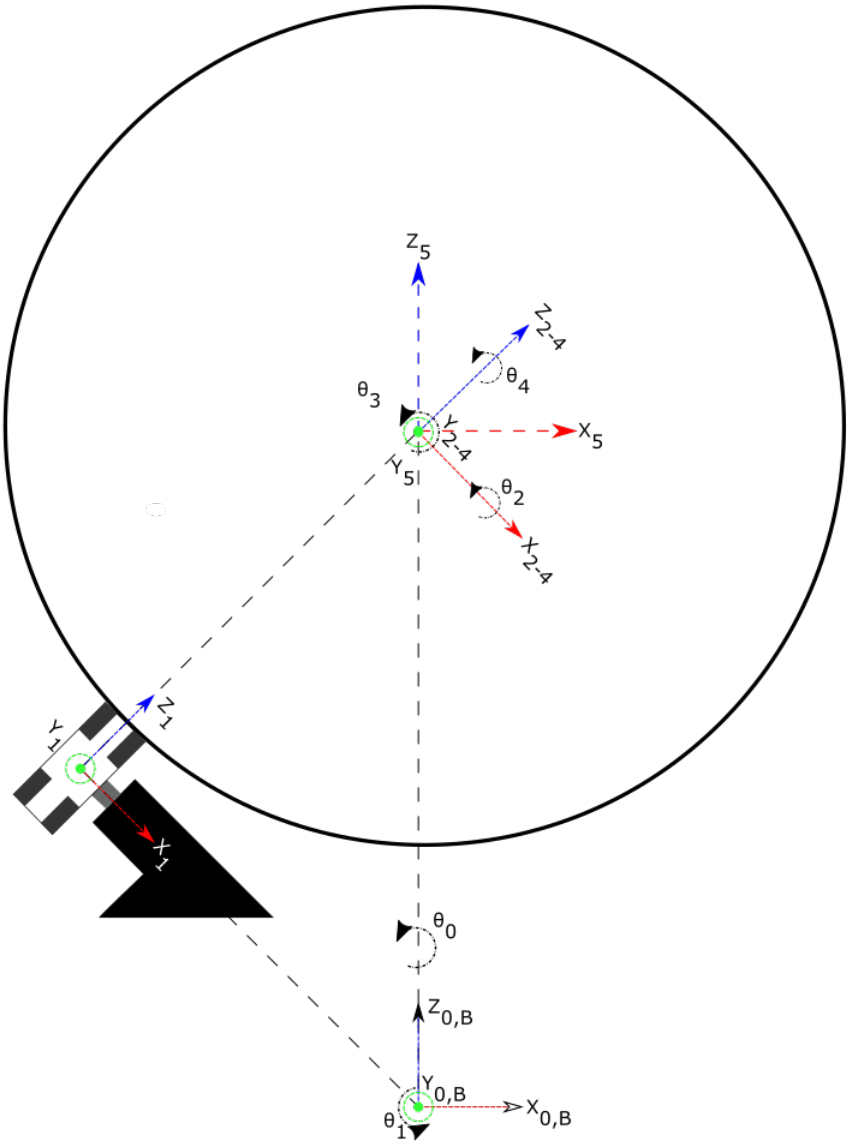


Figure 4.2: Diagram of reference frames for rotation of one equivalent manipulator

Another way to visualize the mechanism is to draw links between the axis and create such an arm as in Figure 4.3. Where the orange base represents the rotations that place the motors in their initial configurations, group $(R_z^{01}R_y^{12})$. The green, yellow and blue arms represent the links in between θ_2, θ_3 & θ_4, and form the mechanism described by $(R_x^{23}R_y^{34}R_z^{45})$. The checkered sphere in the middle represents the center of the fifth reference frame, while the triad extending from it represent the return rotations to this common frame, $((R_y^{12})^T(R_z^{01})^T)$. These extensions from the sphere are then clearly spaced out by 120 degrees around local Z and at 45 degrees about Y from vertical in their respective planes.

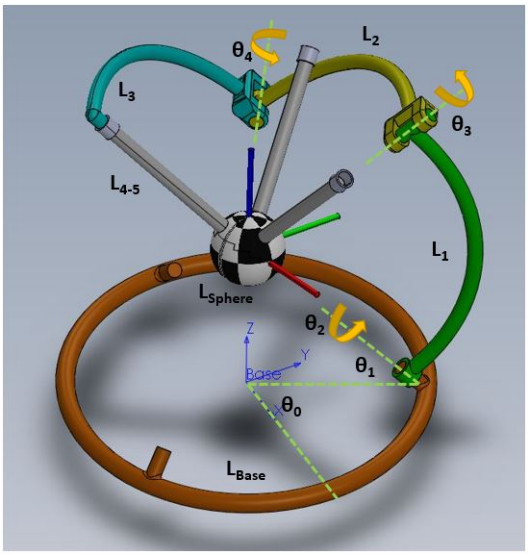


Figure 4.3: One equivalent manipulator arm contributing to the orientation of frame 5

For clarity, the manipulators are discussed individually, however, the complete mechanism looks as it does in Figure 4.1.

So far, the forward kinematics of the mechanism have been established which give the final position of frame 5 if the joint angles, θ_2, θ_3 & θ_4, are known. For Nyku's control purposes, we know the final goal which is our desired orientation of frame 5, the sphere, and the fixed angles of the mechanism. Now the forward kinematics are used to find an expression for the variable joint angles that result in our desired orientation. To do this, we begin by establishing a goal rotation, G, and setting it equal to the forward kinematics, R^{06}.

$$G = R_z(\theta_y)R_y(\theta_p)R_x(\theta_r) \tag{4.3}$$

Notice, that here G is defined in with fixed reference frame conventions such that the desired orientation of the final ball can be given more intuitively in standard roll (θ_r), pitch (θ_p), and yaw (θ_y) angles. As such, the rotations are taken in the order of Z-Y-X for pre-multiplication with the home frame.

Notice that the subscript i is now added to the rotations indicating which arm, A, B, or C, the forward kinematics are computed for.

$$G = R_i^{06} = (R_{zi}^{01} R_y^{12})(R_{xi}^{23} R_{yi}^{34} R_{zi}^{45})((R_y^{12})^T (R_{zi}^{01})^T) \tag{4.4}$$

From here, we isolate the 3 DOF manipulator from the right hand side of the equation. For simplicity, we redefine the manipulator as M_i. Where, each rotation R_i is a function of its joint angle θ_i.

$$M_i = R_{xi}^{23}(\theta_{2i}) R_{yi}^{34}(\theta_{3i}) R_{zi}^{45}(\theta_{4i}) \tag{4.5}$$

Here, the rotations on the right in Equation 4.5 are all with respect to the angles of the manipulator in question. This rotation is taken in the order $X - Y - Z$ as we describe the manipulator from the base in the moving reference frame convention.

$$G = (R_{zi}^{01} R_y^{12}) M_i ((R_y^{12})^T (R_{zi}^{01})^T) \tag{4.6}$$

$$M_i = ((R_y^{12})^T (R_{zi}^{01})^T) G (R_{zi}^{01} R_y^{12}) \tag{4.7}$$

The manipulator indexing i is dropped on R_y^{12} as this rotation is common to all manipulators as is the goal G. The identifying i is also dropped within the matrix elements for brevity as well, but it should be noted that these are the angles of the individual manipulator. Now, on the left we have M_i which is a 3×3 rotation matrix of the standard form:

$$M_i = \begin{bmatrix} c_3c_4 & -c_3c_4 & s_3 \\ c_2s_3 + s_2s_3c_4 & c_2c_3 - s_2s_3s_4 & -s_2c_3 \\ s_2s_4 - c_2c_4s_3 & s_2c_3 + c_2s_3s_4 & c_2c_3 \end{bmatrix} \tag{4.8}$$

where $s_2 = sin(\theta_1)$ and $c_4 = cos(\theta_4)$ etc.

The right hand side of Equation 4.7 is entirely known and will therefor be a 3×3 numeric rotation matrix which we will denote as G_i, with elements labeled as in Equation 4.9.

$$G_i = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \tag{4.9}$$

Substituting $M_i = G_i$ we have:

$$\begin{bmatrix} c_3c_4 & -c_3c_4 & s_3 \\ c_2s_3 + s_2s_3c_4 & c_2c_3 - s_2s_3s_4 & -s_2c_3 \\ s_2s_4 - c_2c_4s_3 & s_2c_3 + c_2s_3s_4 & c_2c_3 \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \tag{4.10}$$

For the purposes of this solution we will be focusing on only the elements shown in Equation 4.11

$$\begin{bmatrix} c_3c_4 & -c_3c_4 & s_3 \\ \ddots & \ddots & -s_2c_3 \\ \ddots & \ddots & c_2c_3 \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ \ddots & \ddots & r_{23} \\ \ddots & \ddots & r_{33} \end{bmatrix} \tag{4.11}$$

From Equation 4.11 we can make the following simplifications using trigonometric identities to find our joints angles, starting with θ_3:

$$r_{23}^2 + r_{33}^2 = (-s_2c_3)^2 + (c_2c_3)^2 = c_3^2(s_2^2 + c_2^2) = c_3^2$$

$$c_3 = \sqrt{r_{23}^2 + r_{33}^2}$$

$$r_{13} = s_3$$

$$tan(\theta_3) = \frac{s_3}{c_3}$$

$$\theta_3 = aTan2\left(\frac{r_{13}}{\sqrt{r_{23}^2 + r_{33}^2}}\right) \tag{4.12}$$

Where aTan2 is the four quadrant arc tangent that returns the result from 0 to 2π. We now use this value in our calculations of θ_2 and θ_4.

$$\frac{-r_{23}}{r_{33}} = \frac{s_2 c_3}{c_2 c_3}$$
$$tan(\theta_2) = \frac{-r_{23}}{r_{33}} = \frac{s_2}{c_2}$$

Although c_3 simplifies our here, we maintain its sign in order to preserve the functionality of aTan2 in Equation 4.13 below.

$$\theta_2 = aTan2\left(\frac{-r_{23} \times sign[c_3]}{r_{33} \times sign[c_3]}\right) \tag{4.13}$$

Last, $\theta_4$is calculated using a similar procedure to θ_2.

$$\frac{-r_{12}}{r_{11}} = \frac{s_4 c_3}{c_4 c_3}$$
$$tan(\theta_4) = \frac{-r_{12}}{r_{11}} = \frac{s_4}{c_4}$$

$$\theta_4 = aTan2\left(\frac{-r_{12} \times sign[c_3]}{r_{11} \times sign[c_3]}\right) \tag{4.14}$$

In Equation 4.15, the procedure for calculation is organized for clarity and re annotated for individual manipulators.

$$\begin{aligned} \theta_{3i} &= aTan2\left(\frac{r_{13i}}{\sqrt{r_{23i}^2 + r_{33i}^2}}\right) \\ \theta_{2i} &= aTan2\left(\frac{-r_{23i} \times sign[c_3 i]}{r_{33i} \times sign[c_3 i]}\right) \\ \theta_{4i} &= aTan2\left(\frac{-r_{12i} \times sign[c_3 i]}{r_{11i} \times sign[c_3 i]}\right) \end{aligned} \tag{4.15}$$

The procedure is then followed for each manipulator, A, B, and C such that the solution matrix in Equation 4.16 can be assembled. In this matrix the top row represents the controllable motor angles before applying the gear ratio from sphere to omni-wheel.

$$Solution = \begin{bmatrix} \theta_{2A} & \theta_{2B} & \theta_{2C} \\ \theta_{3A} & \theta_{3B} & \theta_{3C} \\ \theta_{4A} & \theta_{4B} & \theta_{4C} \end{bmatrix} \tag{4.16}$$

In order to verify that this solution is mathematically correct, the goal position is set to a specific orientation as defined by θ_r, θ_p & θ_y, and then

the resulting joint angles are found for each equivalent manipulator. These joint angles are then used in the forward kinematic calculation resulting in the manipulator pose. the manipulator pose is then subtracted from the goal. If correct the difference matrix, D, should be all 0. This procedure is described mathematically below.

$$D_i = G - R_i^{06}(\theta_{2i}, \theta_{3i}, \theta_{4i}) \tag{4.17}$$

These calculations are performed in MATLAB, the code for which can be found in Appendix B. Using Equation 4.17 as verification of the kinematics, the maximum element-wise difference was -1.4×10^{-17}. As this number is practically 0, it is assumed that the kinematics are valid and that any error is caused by the calculation of trigonometric functions by the computer.

The kinematics was also done using a fixed reference frame convention, the process to which is identical except the following changes.

$$R_{fixed}^{06} = ((R_y^{12})^T (R_z^{01})^T)(R_z^{45} R_y^{34} R_x^{23})(R_z^{01} R_y^{12}) \tag{4.18}$$

$$M_{i_{fixed}} = R_{zi}^{45}(\theta_{4i}) R_{yi}^{34}(\theta_{3i}) R_{xi}^{23}(\theta_{2i}) \tag{4.19}$$

$$\theta_{3i_{fixed}} = aTan2\left(\frac{-r_{31i}}{\sqrt{r_{11i}^2 + r_{21i}^2}}\right)$$
$$\theta_{2i_{fixed}} = aTan2\left(\frac{r_{32i} \times sign[c_3 i]}{r_{33i} \times sign[c_3 i]}\right) \tag{4.20}$$
$$\theta_{4i_{fixed}} = aTan2\left(\frac{r_{21i} \times sign[c_3 i]}{r_{11i} \times sign[c_3 i]}\right)$$

When tested with Equation 4.17, the fixed frame equations also result in a zero matrix. However, these equations do not produce the same results as the equations for the moving reference frame, which had already been developed and tested when the fixed frame equations were tested. These equations require more analysis as there results give very similar results for motor angles. When compared to joint angles resulting from the moving frame calculations, $\theta_{2C_{fixed}}$ is numerically close to $\theta_{2C_{moving}}$, however for $\theta_{2B_{fixed}}$ and $\theta_{2B_{fixed}}$, the fixed frame calculations are clearly twice the value of those done in moving frame. For joint angles on arm A, $\theta_{2A_{fixed}}$ appears to be in the opposite quadrant of $\theta_{2A_{moving}}$.

4.4 Testing Procedure for Equivalent Manipulator Kinematics

In order to validate the inverse kinematics developed in the previous section, the equations had to be tested on the mechanism itself. This was

done by implementing the inverse kinematics in a C++ library, as illustrated in Figure 4.4. In this inverse kinematics library, the inputs are fed in as a vector of desired RPY angles in radians, where they are then used to form the terms given on the right hand side of calculations outlined in Equation 4.15. In practice, the analytical formula for each of the terms in these equations is found by symbolic substitution in MATLAB, such that the terms in G_i are represented by long trigonometric expressions dependent only on the input RPY angles. These expressions are not produced here fore brevity. However, they be found in the GitHub repository provided in Appendix A. This is library is then used to create the test program in the ROS architecture as described in more detail the Chapter 3. Additionally, this test is conducted with the ribs, hardware layers and neck removed from Nyku, leaving only the empty hemisphere on the Omni-wheel Base.

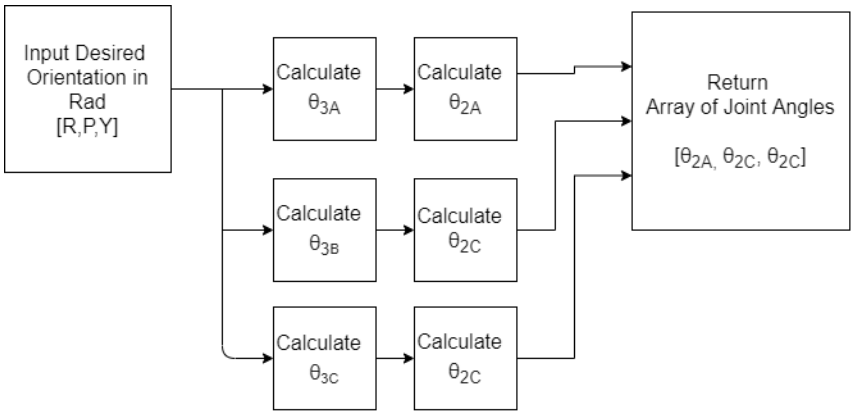


Figure 4.4: Flow Chart for IK Calculation

The test program takes a CSV file containing a list of desired orientations, and parses each row into a 3×1 goal orientation vector. This goal orientation vector is then fed through the inverse kinematics class and returns the three goal motor positions which are then sent to the PID controller on the ROS enabled Teensy4.0 motor controller. The test program then waits for the position error of the motors to be 0, and then waits for half a second, to guarantee stable IMU readings, before taking a quaternion reading, converting it, and recording the resultant RPY of the hemisphere in a CSV created when each test is run. A detailed flow chart for this program can be found in Figure 3.33.

In order for this test to succeed, the IMU, in this case a Razor 9DOF IMU from Sparkfun, must be calibrated according to the manufacturers instructions. Next, the coordinate system of the IMU must be aligned with the coordinate system of the Omni-Wheel base. This is achieved by outputting the IMU orientation so that the orientation can be verified and then mounting the IMU in the correct position on the Hemisphere, which has a known coordinate system. The test setup can be seen in Figure 4.5. Next the system is powered on and the test program is run with the corresponding CSV for the desired test. Once the test is complete the data is graphed with the goal RPY and the actual RPY of the mechanism on the same axis. Error between goal and

Figure 4.5: Test setup for inverse kinematics validation

actual orientation was calculated for each point with respect to Pi in order to avoid division issues when the goal position was 0. This data is also plotted

$$error = \left(\frac{goal_\theta - actual_\theta}{\pi} \right) * 100 \tag{4.21}$$

When graphing the data for yaw, the initial measured yaw is subtracted from all yaw data points and is labeled as the yaw offset on the graphs. This is done as the yaw measurements from the IMU experience large drift according to the manufacturer, meaning that the alignment initially done during mount-

ing is not reliable. By subtracting the initial yaw, we are able to zero the IMU about that axis for each test.

Tests were developed for roll, pitch, and yaw motions individually as well as one combined test that sent a series of motions about all axes to the mechanism. The individual tests each had 199 goal orientations, and the combined test had 121 goal orientations. These tests were limited to oscillations between $\pm 30^\circ$ about any axis. This was the limit of motion of the hemisphere so that its equator would not dip below the chord height projected onto the sphere by the omni-wheels. Yaw could have been tested with larger angles however the data would not have been as comparable between the axes. The combined test is intended to show that each axis of the goal orientation is independently controlled. The individual axis tests are intended to measure the repeatability of the system and detect any steady state error that may occur.

4.5 Omni-wheel Inverse Kinematics Testing

Results

The results of the testing described in the previous section are presented and discussed here. Below, the graphs compiled from those tests are shown.

In Figure 4.6 the test for combined motion graphed. Here the Omni-wheel base oscillates the full range of motion about the x axis, follow by the y axis and finally the z axis. The actual position is found to trace the goal position well, except for note-able spikes in error for yaw when the negative pitch angle is set as the goal position. For roll and pitch angles, there is a maximum error of -8%, however, there is a spike of 11% in the yaw for the negative pitch angle, visible in Figure 4.7. The next tests are focused on the repeatability of reaching target rotations about a singular axis.

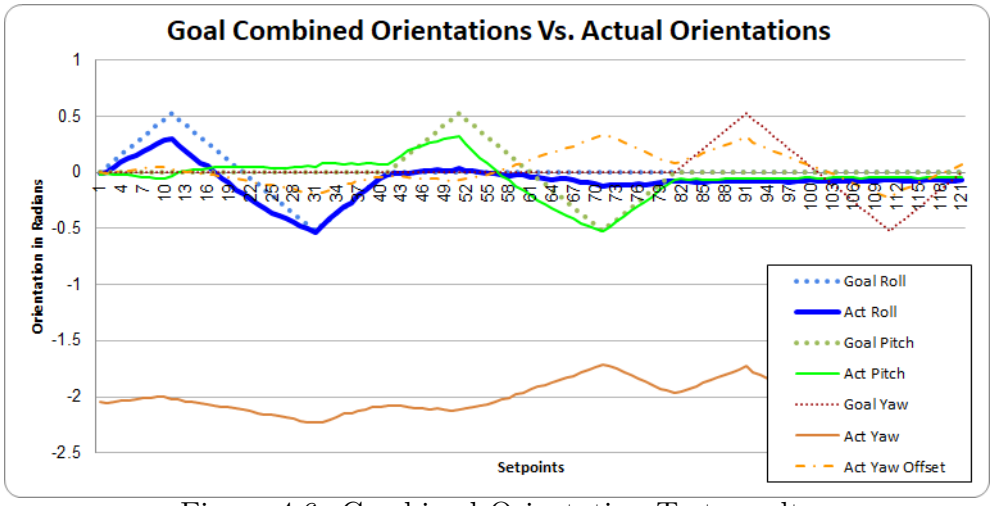


Figure 4.6: Combined Orientation Test results

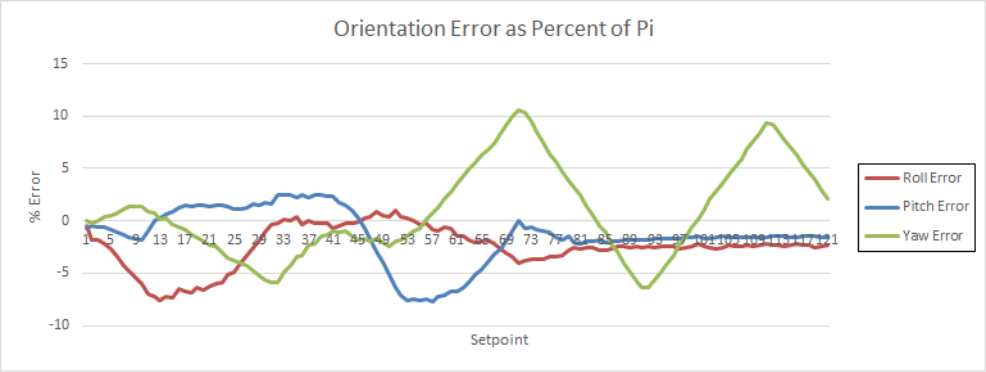


Figure 4.7: Combined Orientation Test Error

The repeatability tests are run in order; roll, pitch, yaw. Each test consists of four and a half oscillations in the range of motion and a stream of 20 goal orientations at 0 rotation where cumulative error can be examine. When looking at the pitch and roll tests in Figure 4.9 and Figure 4.8, it can be seen that the mechanism only reaches the goal position for negative goal rotations.

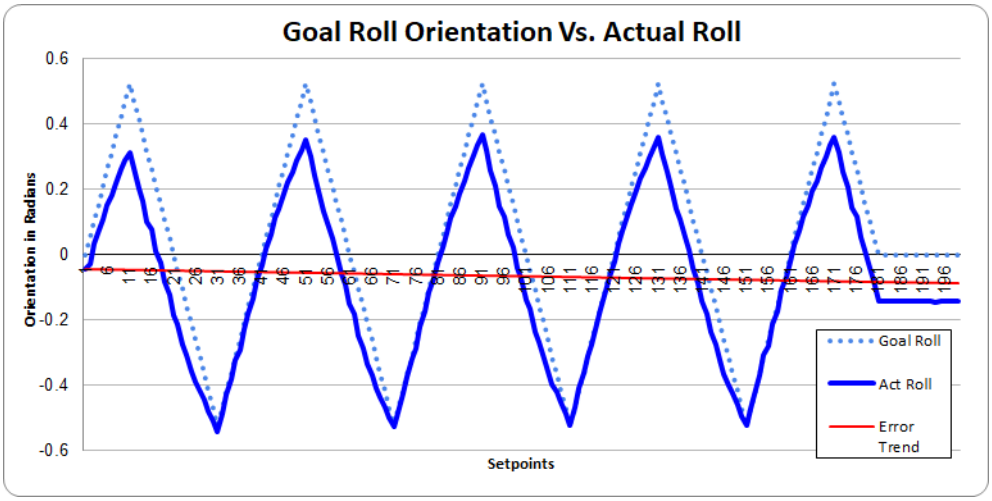


Figure 4.8: Roll Orientation Test results

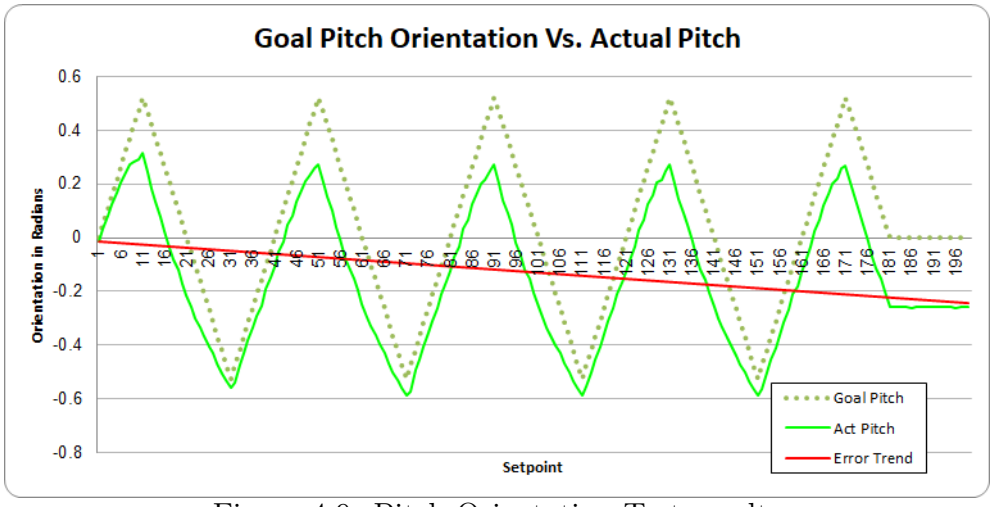


Figure 4.9: Pitch Orientation Test results

For positive roll goal rotations, there is a maximum error of -7% and for pitch there is a maximum error of -9% also occurring at negative goal orientations. When examining the error graphs for pitch and roll , Figure 4.10 & Figure 4.11, toward the end of the trajectory it is observed that there is a static error of around -4.5% for roll orientations and -8% for pitch orientations. The trend line of this offset error is graphed on the orientation graphs as well.

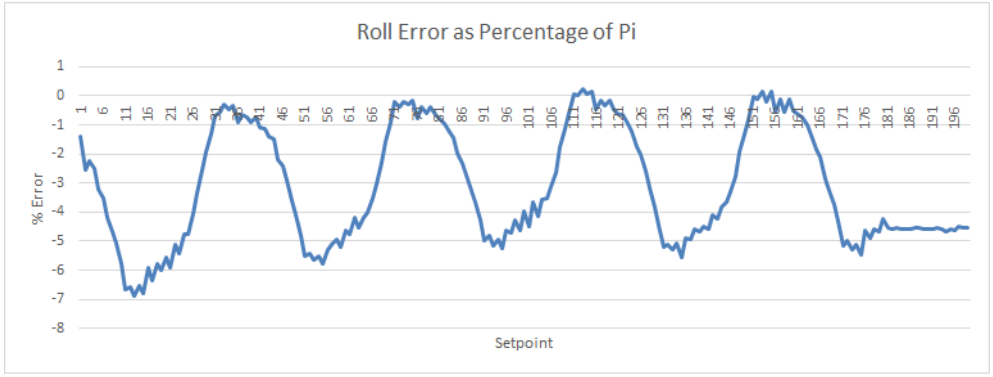


Figure 4.10: Roll Orientation Test Error

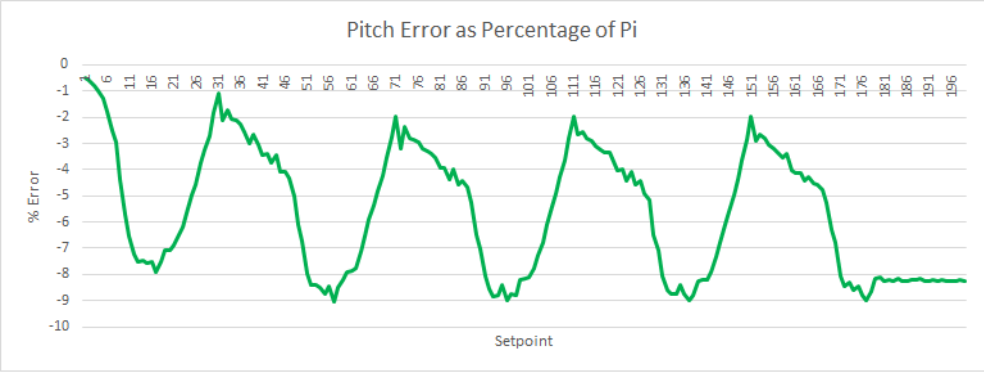


Figure 4.11: Pitch Orientation Test Error

When the data for yaw orientations is analyzed, Figure 4.12, it is observed that the goal position in yaw is never reached, but instead that there is a consistent gap between the actual and goal rotations. This is observed in the error plots, Figure 4.13, as an oscillation in position error between ±8% corresponding to negative and positive set points. When the 0 rotation goal stream is sent at the end of the test, there is no cumulative static error indicating that the system returned to home position.

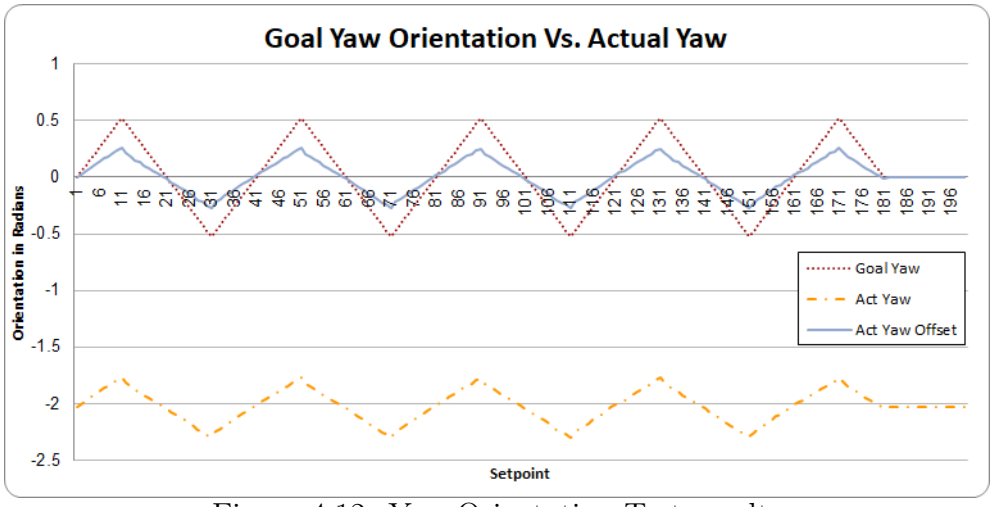


Figure 4.12: Yaw Orientation Test results

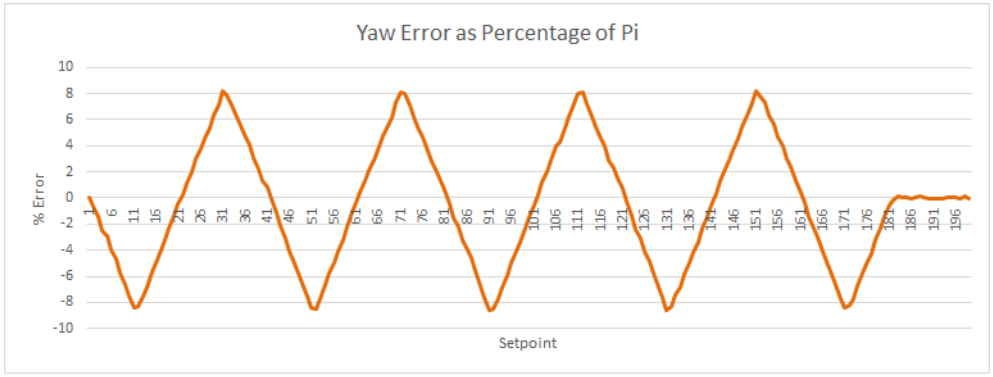


Figure 4.13: Yaw Orientation Test Error

Discussion of Testing Results

The testing conducted informs us about the performance of the system and also suggests some improvements that can be made to both the control architecture and the hardware. From the tests for roll and pitch orientations, it is tempting to say that the kinematic equation models the system poorly

when the goal orientation angles are positive. However, when the yaw data is observed, it is obvious that the amplitude of the actual position does not match the amplitude of the goal position, and that the error between the two increases in direct proportion to the magnitude of rotation angle increase. When this observation is noted, it also becomes apparent in the rotations about X and Y. The difference being that rotations about X and Y oscillate about their accumulated position error as evidenced by the error graphs in Figure 4.10 and Figure 4.11. Over 4.5 oscillations, roll motions accrued an error of -4.5%, while the pitch motions had a cumulative error of -8%. Although, it must be noted that the roll and pitch orientations start with error due to initial hemisphere positioning. About X this error is -1.4% and about Y it is 0.47%, however, this does not explain the rapid accumulation of error. Yaw rotations do not show an initial error due to the offset method described in the procedures. Nor do these rotations demonstrate an accrued error over the testing period. In all cases, the error peaks at the extremes of the goal orientation trajectories. In yaw this is clear, but in pitch and roll is only apparent when the oscillations are observed over the error trend line in Figure 4.8 and Figure 4.9. If these error trend lines are taken as the reference point of the oscillations then the deficit in amplitude is more clear.

When observing the tests visually it is clear that the cumulative error present in roll and pitch motions is due to slip between the omni-wheels and the hemisphere. Because the ballast in the sphere is not mounted in the center, but slightly lower, the center of gravity of the sphere changes in certain orientations. As the mechanism approaches the positive extremes of the trajectory the center of mass of the ball is furthest away from the motors that control its position the most. This causes slip on the omni-wheel that influences the orientation the most, resulting in significant undershoot. When returning to 0 and heading toward negative orientation angles, this undershoot compensates for the smaller amplitude of the motion, causing the error for negative positions to be very low. At negative positions the mass of the ballast is over the most influential motor and does not slip out of static friction. In yaw, this is not an issue as the center of mass of the hemisphere does not change. The Atlas sphere designed by the team at Carleton University, solves the slipping issue by using a set of roller bearing corresponding to the omni-wheel orientation above the sphere which is compressed with a screw to increase the normal force on the omni-wheels and ensure that the force remains equal throughout the motion [7] [28] [44]. For Nyku, this configuration does not work as the omni-wheel interface is only a hemisphere and top is reserved for the body.

This slip, however, does not explain the undershoot caused by the deficit in actual orientation amplitude.

Although yaw is the best example (Figure 4.13), motions about all axes show a direct linear relationship between goal orientation magnitude and error magnitude. This could be do to a difference between the theoretical gear ratio and actual mechanical advantage of the omni-wheels on the sphere. Although this ratio is calculated based on the designed diameter of the hemisphere and the diameter of the omni-wheels as provided by the manufacturer, actual sizing may be slightly different causing the slope of the actual orientation graph to be different than the goal orientation plot's. Another reason for this possible difference in gear ration is the nature of the omni-wheel configuration. Because the rollers on each side of the wheel are 45° out phase, the omni wheel can have a slightly lower diameter if the tangency point of the omni-wheel on the hemisphere is between rollers.

Even though the linear relationship between set orientation and error indicates an issue in gear ratio, there may also be an issue with the elevation angle of the motors as modeled in the kinematic equations. Although the motor mounts are designed to sit at an elevation angle of 45°, the actual 3d printed parts may be slightly out of tolerance. Another issue is that when the weight of the hemisphere is placed on the omni-wheel interface it can flex

the 3d printed motor mounts as well as the acrylic base. The base flex is mitigated by an aluminum reinforcement plate, but the motor mounts may still be flexing out of tolerance. This flexing decreases the elevation angle of the motor mounts and changes the orientation of the principle axes for the equivalent manipulator on each wheel.

Despite errors in position amplitude at extreme angles and the issues with non-constant normal force on the omni-wheels, it is clear that the kinematic model used is a good representation of the system. The actual orientation is found to track the goal orientation well, given that the maximum error magnitude for any test was 10% which occurred during the combined motion. It hypothesized that this spike in error could be caused by the mounting position of the IMU which is slightly above the center of the sphere, so that when pitch is applied there may be an induced yaw in the elevated plane of the IMU.

Never the less, the issues made apparent by the testing also indicate what solutions could be used to mitigate them. In order to change the slope of the actual orientation graph, it is possible to increase the gear ration mapping the inverse kinematics to the motors, which would presumably increase the magnitude of the output rotation. Secondly, to counter act the slip occurring in the system the IMU could be introduced into the control loop. By comparing goal rotation to actual rotation in the control loop this cumulative error can

be identified and corrected. If the difference between the goal orientation and the actual orientation is about a certain threshold, the inverse kinematics of the system can be run using the actual position to extract the corresponding joint angles that would result in the actual position. These joint angles can then be used to adjust the encoder values on the motor controller itself. The motion can also be limited to angles that do not cause as much slip, but this is not as preferred.

The testing conducted demonstrates the viability of the parallel equivalent manipulator kinematics developed for control of the Omni-wheel Base. From the data we can see reasonable position tracking, and quantitatively the motions are expected when observed visually. However, due to the architecture of the system design for use in Nyku and not to optimize testing of the kinematics, hardware testing was limited to motions not exceeding the hemisphere forming the bottom of Nyku's body. Future testing of the kinematics should be done with a whole sphere with a internally mounted, battery powered, IMU. Orientation could also be measured using a camera and marker system as in the research done by Beranek and Hayes on the orientation of their Atlas sphere [7]. This would allow for rotations which invert the sphere to test the validity of the kinematics when the goal orientations are greater than 30°.

Chapter 5

Conclusion

5.1 Future Work

As is the nature of any design, the initial work lays the foundation for future improvements until all the design goals are met. At this stage in Nyku's development, the next steps are clear. So far, the individual subsystems are all tested and have been mechanically and electrically refined. The control schemes have been coded and well documented with intuitive libraries to use. What follows is the orchestration of the individual systems into a combined system that performs the tasks required by the studies to be conducted. During this orchestration, certain parts may be improved upon to optimize performance. What follows is a discussion of the work that still needs to be done so that the next phases of research may be conducted and what the next phases of research may look like.

5.1.1 Aesthetic Refinements

First, it must be addressed that the outer covering of Nyku can be made more aesthetically appealing. For the current version of Nyku, the outer covering was sewed by a novice and is therefor sub-optimal. If the outer covering is designed and manufactured by someone with more experience the result could be far better. Because Nyku is designed to be open source hardware, this sewing pattern designed could be made by the individuals building Nyku, such that each outer covering is unique. Currently, the documentation for the outer covering is comprehensive enough to produce a covering which fits the robot, from here an experienced crafts person could easily optimize the sewing pattern. Secondly, the eyes may be optimized for each user, which also requires some additional aesthetic consultation, however the source code is provided and can be easily edited, so this should not prove to be an issue. In addition, eye shape could be coded for different emotions such as frustration, happiness, sadness, etc.

5.1.2 Electrical Refinement

In order to perfect Nyku, the wiring should be redone and optimized. Currently, some of the connections in the body are done with very long cables

which were used due to rapid accessibility, as they were already in the laboratory. In order to more easily debug and to improve the professionalism of the system, these cables could be purchased at the correct size and replaced. Another improvement that should be made is that the PCB for the motor controller circuit in the Omni-wheel Base could be optimized for reliability as well as including the motor driver circuits. In this way, the ROS enabled motor controller could be far smaller and not require an additional mounting before testing. As discussed in Chapter 3, the motor controller circuitry is implemented on a perforated board which still requires packaging before tests can be conducted with ASD children. By using an integrated circuit to both read the encoders and control the motors, this system could be tucked into the Omni-wheel base itself.

5.1.3 Code Maintenance and Refinement

Currently, the control code is implemented on a Linux PC as speed of compilation is greatly limited on the Raspberry Pi, severely increasing development time. In order for the system to be fully integrated within Nyku, the code must be migrated over to the Raspberry Pi itself. While this is being done, it is highly suggested that the operating system of Nyku be upgraded

to the most up to date, supported, system. At the time of writing, this is Ubuntu Mate 18 for ARM running ROS Melodic, currently there are pre-built images of this system, so this task should prove easy. From here, the packages can be downloaded from the GitHub provided and the catkin workspace compiled. Again, as noted in Chapter 3, these packages provide the interfaces for control of the system but do not yet perform the tasks necessary to conduct the RAT studies (proposed below). The last step in implementing Nyku as a therapeutic tool is to write a policy maker code designed to fulfil the tasks outlined in the Engagement Plan detailed below. But of course, this can only be done after the plan has been vetted.

5.1.4 Engagement Plan for testing in ASD Children

Most importantly, Nyku must be put into practice as a therapeutic tool and evaluated. The intended engagement plan is presented here for evaluation so that it can be optimized before implementation.

To begin, a number of high functioning ASD children will be selected based on uniform criteria from a pool available to the DU Computer Vision & Robotics Lab. These children would be asked to engage with Nyku while playing Minecraft, which serves as a joint attention task. Video games are

selected as the joint attention task as they have been shown to be of high interest to high functioning ASD children due to their repetitive and predictable nature while providing the mechanisms for positive and negative reinforcement [38]. Additionally, video game therapy has shown positive outcomes for ASD children. In a study conducted in Spain "the emotions evoked by the game were positively linked with higher physiological reactivity in the participants, they exhibited new coping styles to handle real-life stress, and they exercised greater self-control strategies when confronted with impulse-triggering situations [11]."

Moreover, Minecraft was selected as the game because it is widely accepted in society and provides age appropriate content while still allowing for conflict. Minecraft is an open world game where the player is able to create things in their environment using blocks that they mine from resources on the randomly spawned map. The game provides a creative mode and a survival mode allowing for variation in game play and therefore therapy dynamic. This game provides the creativity of art therapy while allowing the child and robot to develop their own narrative. By establishing their own narrative, we aim to increase the theory of mind response from the child with regards to the robot to aid in team building skills. This is intended as a joint attention game of high complexity where the Nyku will be able to meaningfully interact with the

child in various scenarios. In the games creative mode, Nyku will be able to give suggestions about what to build and where; if the child seems stuck Nyku can nudge the child to more activity as well as encouraging the child when they are playing creatively. During the games survival mode, Nyku can react by mimicking the child's attentive posture as well as recoiling in an averse affect when a negative stimuli occurs. In creative mode, the pace is slower and the child can triadically interact with Nyku and the game. In survival mode, the child can observe how Nyku reacts to negative stimuli and associate those responses. The conflict moments in the game are used as teachable moments with Nyku, where it can say things like "Wow, that was very frustrating! When I get frustrated I usually react like this.", then Nyku would assume a frustrated posture and eye expression.

For each subject, a new campaign would be started and the progress would pick up where it left off for each session. Given the popularity of Minecraft, and the incentive to continue a game, it is possible that participant enthusiasm could increase. During each session, Nyku would subtly mimic the posture of the child in a graded manner. In the first session, Nyku would mimic the posture 10% progressing slowly to 100% posture mimicry for the last session. Each session would be recorded and the amount of utterances of the child would be measured and categorized by type; child-Nyku, child-game,

child-nyku-game, etc. After each session, the child would also be asked to fill a survey aimed at measuring the effectiveness of Nyku. This survey would be targeted at finding out if Nyku is eliciting theory of mind responses from the child and would include questions such as:

1. Do you think Nyku enjoys Minecraft?
2. Do you think Nyku was frustrated?
3. Do you think Nyku had fun playing with you?

This information, combined with parent interviews, would create a set of data measuring Nyku's effectiveness as a therapeutic tool as well as indicating the changes that need to be made in the design to improve its efficacy.

5.2 Contributions

The results of this thesis are two fold. Firstly, the robot, Nyku, was specifically designed around the needs of ASD children with the goal of facilitating non-verbal communication skills with robot-assisted therapy. This was done by simplifying the bandwidth of non-verbal communication inherently available. As such, Nyku was designed to be capable of posture mimicry while serving as a therapeutic tool.

Secondly, in the pursuit of this posture mimicry system, a novel solution to the inverse kinematics of omni-wheel position control of a spherical body

was developed. By modeling the omni-wheel hemisphere interface in the Omni-wheel Base as three serial link parallel manipulators a one of a kind analytic solution to the problem was found.

As such, a non-provisional US patent was filed for Nyku's design. The technical work will also be submitted to the International Conference on Robotics and Automation (ICRA 2021) among other publications and conferences.

5.3 Discussion

Over the course of two and a half years, Nyku was developed and that work is presented in this thesis. This work started with the development of a set of requirements specific to the needs of ASD children, which was then used to design Nyku. These designs were implemented through a combination of mechanical, electrical, and computer engineering. From these designs a novel solution for the control of the Omni-wheel Base was found.

By examining the needs of ASD children, and the work done in HRI and RAT to provide therapeutic tools for the populations Nyku's design goals were identified, as presented in the beginning of Chapter 3. In short, it was necessary to create a robot capable of simple NVC, by mimicking body posture, while not eliciting anxious behaviors from the ASD children the robot is

intended to serve. Naturally, the robot also had to appeal to children with ASD while being able to serve as a therapeutic avatar for an experienced therapists to include as a part their treatment plans. Given these goals, the engineering goals were dictated.

In order to meet these requirements, Nyku was built to resemble a penguin whose form is capable of avoiding Mori's Uncanny Valley, while appealing to children due to its furry outer body and large size. Nyku was also designed without a mouth, allowing the eyes to be the central feature of the face. This was done to reduce the NVC communication bandwidth with the intent of reducing possible anxiety inducing stressors. This zoomorphic embodiment lent itself to the hemispherical body stem that allows for the pose control utilized in posture mimicry. In the head, Nyku contains a camera and a microphone providing hardware capability to stream audio and video to a therapist controlling the robot in another room.

These mechanical systems were built, tested and refined. Starting with the neck & head and ending with the Omni-wheel base, each component was designed to meet the needs of children with ASD. The head was designed to omit the mouth and to be aesthetically pleasing. The neck is then designed to move the head with smooth natural motion. From there the body design is adapted to fit the changes of neck. And the cycle repeats for each component.

In order to meet these requirements, manufacturing techniques like sewing are used along side traditional machining, laser cutting, and 3D printing. To date, the current version of Nyku has been through two design iterations.

To test the components, the electrical systems powering the assemblies, as well as the circuits for control, underwent their own development processes. This includes the power supplies of both Nyku & its Omni-wheel base as well as the control systems for the neck, flaps, microphone and speaker all were designed, created, tested, and refined. In addition to this, a closed loop feedback system was created for the position control of DC omni-wheel motors was designed and implemented on a custom PCB, despite issues mentioned in Chapter 3.

In addition to creating the physical systems, the computer systems were also engineered. The Raspberry Pi operating system was compiled to accept standard robotic system libraries, such as ROS and OpenCV. And after the configuration was done, the system control code as written for the Omni-wheel base and neck in C++. A GUI to display the eyes was also created in Python. This work includes libraries implementing the novel kinematic equations for the orientation of a sphere. It also includes writing a ROS enabled motor controller on a Teensy4.0.

5.4 Concluding Thoughts

Using the principles of engineering design, a social robot for children with autism was successfully created. First, the design requirements were drawn from the needs of the desired user population, these design requirements were then translated into a full design. And finally the design was implemented, tested, and refined. This design process was well detailed and documented for posterity. During this process the novel kinematics for the control of the pose orientation system in the Omni-wheel Base was discovered, and subsequently tested & evaluated. As a result, Nyku is now a fully functional social robot physically capable of posture mimicry in torso & neck, eye gaze, speech, and audio recording. The success of this design did not come easily, as indicated by the many iterations each system underwent. By learning mechanical, electrical, and computer engineering disciplines in tandem, their inter-connectivity became very evident, as did a need for a strong understanding of all foundations.

Due to COVID-19, the system was completed and tested outside the laboratory and such the scope of this work was limited to the design, construction, and testing of the system. Under normal circumstances, the system would have been used to conduct studies using ASD subjects so that the goals

of the design could be evaluated. Despite this limitation, this thesis represents a multi-disciplinary effort to design a robot for ASD children. As such, this work serves as the basis for any future work on Nyku. The design and manufacture process are comprehensively documented so that Nyku may be reproduced and improved upon. This includes mechanical, electrical and computer engineering work; the three core engineering disciplines that form the basis for mechatronics.

The nature of robotic development is three pronged. The mechanical systems cannot be tested if the motors are not powered, and the motors cannot be controlled without programming. Because of this, the study of robotics stands on the three pillars of mechanical, electrical, and computer engineering. During the design process, these elements need to be considered in tandem. One cannot pick a motor without knowing if it meets there mechanical needs, the power it will take to run it, and how its feedback system will be used in its control. Therefore, the success of a robotic system then depends not on an interest in one particular field, but on a passion for building, creating, & problem solving in all its forms.

Appendix A

A.1 Torso Components

- USB Speakers
- 3X Buck converter 5amp max
- Buck converter 12amp max
- Raspberry Pi 4B+ (ROS Melodic on Debian Buster)
- Pololu Maestro 12 Channel Servo Controller
- Anker 7-Port Powered USB 3.0 Data Hub
- Intel Movidius Neural Compute Stick
- 10 Amp power switch
- Turnigy 4S 14.8V 6.6mAh battery with XT90 connector
- 2X HiTech HS-5087MH Digital Servo

A.2 Omni-Wheel Base Components

- 3X Pololu Encoded Gear Motor #4867
- Raspberry Pi 3B+ (ROS Melodic on Debian Buster)

- Teensy 4.0 Micro Controller
- Cytron FD04A Motor Driver
- 12V 15A Power Supply
- Buck Converter to power Raspberry Pi from Power Supply
- DuPont terminated cables for direct motor interface
- Jumper Cables for motor controller/driver interface
- 2X USB B Cable

A.3 Neck Components

- 3X Dynamixel AX-12A Servo Motors
- SMPS2 Dynamixel Power Adapter
- U2D2 Dynamixel USB Adapter
- M4 Dog-point Screws (Misumi CBBG4-16-RC)
- M4 Heat Set threaded inserts (McMaster 94459A150)
- 4X 3x6x2mm Bearings (VXB KIT9059)

A.4 Head Components

- Pololu Maestro 12 Servo Controller
- ELECROW 5 Inch Touch Screen 800x480 TFT LCD
- ReSpeaker Mic Array v2.0

- Orange LED 5mm
- Raspberry Pi Spy Camera
- Flex Cable extension for camera (Adafruit 2087)
- Flex Cable connector (Adafruit 3671)

A.5 Miscellaneous Parts for Construction

- 6-32 Heat Thread inserts (McMaster 93365A130)
- 6-32 Bolts Nuts Washers
- 1.75mm PLA filament for 3D printer (Clear and Black)
- 1.75mm TPU filament for 3D printer (Black)
- Acrylic Sheet Various Thickness
- Various Perf-boards for semi-permanent prototyping of circuits
- DuPont cables
- 24 gauge wire, various colors

A.6 Tools Used

- Upgraded Ender 3 3D Printer
- 24"x48" CO2 Laser Cutter
- Stratasys Fortus 450MC Printer
- CNC mill

- Manual Lathe

A.7 Design Programs Used

- SolidWorks
- Eagle PCB Design
- VS Code for Linux with PlatformIO package

A.8 GitHub Sources

1. dynamixel_control_hw - `https://github.com/resibots/dynamixel_control_hw`
2. neck_dyna_test - `https://github.com/dstoianovici/neck_dyna_test`
3. Teensy4MotorController - `https://github.com/dstoianovici/Teensy4MotorController`
4. nyku_omniwheel_base - `https://github.com/dstoianovici/nyku_omniwheel_base/`
5. rosserial_arduino - `https://github.com/ros-drivers/rosserial/tree/noetic-devel/rosserial_arduino`

A.9 Abbreviations

1. ASD: Autism Spectrum Disorders
2. CDC: Center for Disease Control
3. DOF: Degree(s) of Freedom
4. DUCV: University of Denver Computer Vision and Robotics Lab
5. HRI: Human Robot Interaction
6. NCS: Neural Compute Stick
7. RAT: Robot-Assisted Therapy
8. RPi: Raspberry Pi
9. RPY: Roll, Pitch, Yaw
10. URDF: Universal Robot Description File
11. YAML: Yet Another Markup Language
12. LQR: Linear Quadratic Regulator

Bibliography

[1] *1 in 88 children : a look into the federal response to rising rates of autism : hearing before the Committee on Oversight and Government Reform, House of Representatives, One Hundred Twelfth Congress, second session, November 29, 2012.* eng.
U.S. Congressional documents. Congressional hearings. 2012.

[2] Hojjat Abdollahi, Ali Mollahosseini, Josh T. Lane, and Mohammad H. Mahoor.
"A pilot study on using an intelligent life-like robot as a companion for elderly individuals with dementia and depression". eng.
In: *2017 IEEE-RAS 17th International Conference on Humanoid Robotics (Humanoids)*. IEEE, 2017, pp. 541–546.
ISBN: 9781538646786.

[3] Joel Aronoff, Barbara A. Woike, and Lester M. Hyman.
"Which are the stimuli in facial displays of anger and happiness? Configurational bases of emotion recognition." In: *Journal of Personality and Social Psychology* 62.6 (1992), pp. 1050 –1066.
ISSN: 0022-3514. URL: http://du.idm.oclc.org/login?url=https://search.ebscohost.com/login.aspx?direct=true&db=pdh&AN=1992-34928-001&site=ehost-live&scope=site.

[4] F. Askari, H. Feng, T. D. Sweeny, and M. H. Mahoor.
"A Pilot Study on Facial Expression Recognition Ability of Autistic Children Using Ryan, A Rear-Projected Humanoid Robot".
In: *2018 27th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*. 2018, pp. 790–795.

[5] Marian R. Banks, Lisa M. Willoughby, and William A. Banks.
"Animal-Assisted Therapy and Loneliness in Nursing Homes: Use of Robotic versus Living Dogs". In: *Journal of the American Medical Directors Association* 9.3 (2008), pp. 173 –177. ISSN: 1525-8610.
DOI: https://doi.org/10.1016/j.jamda.2007.11.007.
URL: http://www.sciencedirect.com/science/article/pii/S1525861007005166.

[6] Janet B. Bavelas, Alex Black, Charles R. Lemery, and Jennifer Mullett.
“”I show how you feel”: Motor mimicry as a communicative act.” In: *Journal of Personality and Social Psychology* 50.2 (1986), pp. 322–329. DOI: 10.1037/0022-3514.50.2.322.

[7] R. Beranek and M. J. D. Hayes.
“Position Level Kinematics of the Atlas Motion Platform”. eng. In: *Advances in Robot Kinematics: Motion in Man and Machine: Motion in Man and Machine*. Dordrecht: Springer Netherlands, 2010, pp. 385–392. ISBN: 9789048192618.

[8] Randolph Blake, Lauren M. Turner, Moria J. Smoski, Stacie L. Pozdol, and Wendy L. Stone. “Visual Recognition of Biological Motion is Impaired in Children With Autism”.
In: *Psychological Science* 14.2 (2003). PMID: 12661677, pp. 151–157. DOI: 10.1111/1467-9280.01434.
eprint: https://doi.org/10.1111/1467-9280.01434.
URL: https://doi.org/10.1111/1467-9280.01434.

[9] John-John Cabibihan, Hifza Javed, Marcelo Ang, and Sharifah Mariam Aljunied. “Why Robots? A Survey on the Roles and Benefits of Social Robots in the Therapy of Children with Autism”.
In: *International Journal of Social Robotics* 5.4 (2013), pp. 593–618. ISSN: 1875-4805. DOI: 10.1007/s12369-013-0202-2.
URL: https://doi.org/10.1007/s12369-013-0202-2.

[10] D. Catalano, L. Holloway, and E. Mpofu.
“Mental Health Interventions for Parent Carers of Children with Autistic Spectrum Disorder: Practice Guidelines from a Critical Interpretive Synthesis (CIS) Systematic Review”. eng. In: 15.2 (2018).
URL: https://doi.org/10.3390/ijerph15020341.

[11] Hans Concepcion.
“Video Game Therapy as an Intervention for Children With Disabilities: Literature Review and Program Protocol”. eng.
In: *Therapeutic Recreation Journal* 51.3 (2017), pp. 221–228.
ISSN: 00405914.
URL: http://search.proquest.com/docview/1940121394/.

[12] Mark Coulson. "Attributing Emotion to Static Body Postures: Recognition Accuracy, Confusions, and Viewpoint Dependence". In: *Journal of Nonverbal Behavior* 28.2 (2004), pp. 117–139. ISSN: 1573-3653. DOI: 10.1023/B:JONB.0000023655.25550.be. URL: https://doi.org/10.1023/B:JONB.0000023655.25550.be.

[13] John J. Craig. *Introduction to Robotics: Mechanics and Control*. 4th. USA: Addison-Wesley Longman Publishing Co., Inc., 2018. ISBN: 0201095289.

[14] Charles Darwin. *The expression of the emotions in man and animals*. D. Appleton and company, 1905. URL: http://hdl.handle.net/2027/osu.32435064475668.

[15] Geraldine Dawson, Sara Jane Webb, and James Mcpartland. "Understanding the Nature of Face Processing Impairment in Autism: Insights From Behavioral and Electrophysiological Studies". eng. In: *Developmental Neuropsychology* 27.3 (2005), pp. 403–424. ISSN: 8756-5641. URL: http://www.tandfonline.com/doi/abs/10.1207/s15326942dn2703_6.

[16] *Diagnostic and statistical manual of mental disorders : DSM-5.* eng. Fifth edition. 2013. ISBN: 0890425574.

[17] Irma Dosamantes-Beaudry. "Somatic Transference and Countertransference in Psychoanalytic Intersubjective Dance/Movement Therapy". eng. In: *American Journal of Dance Therapy* 29.2 (2007), pp. 73–89. ISSN: 0146-3721.

[18] Mustafa Erden. "Emotional Postures for the Humanoid-Robot Nao". eng. In: *International Journal of Social Robotics* 5.4 (2013), pp. 441–456. ISSN: 1875-4791.

[19] David J. Feil-Seifer and Maja J. Matarić. "Robot-assisted therapy for children with Autism Spectrum Disorders". In: *Conference on Interaction Design for Children: Children with Special Needs*.

Chicago, Il, 2008, pp. 49–52.
URL: http://robotics.usc.edu/publications/588/.

[20] W.F. Flack, J.D. Laird, and L.A. Cavallaro.
"Emotional expression and feeling in schizophrenia: effects of specific expressive behaviors on emotional experiences." eng.
In: *Journal of clinical psychology* 55.1 (1999), pp. 1–20.
ISSN: 0021-9762.
URL: http://search.proquest.com/docview/69657708/.

[21] Marco Franceschini, Michela Goffredo, Sanaz Pournajaf, Stefano Paravati, Maurizio Agosti, Francesco De Pisi, Daniele Galafate, and Federico Posteraro.
"Predictors of activities of daily living outcomes after upper limb robot-assisted therapy in subacute stroke patients". eng.
In: *PLoS ONE* 13.2 (2018), e0193235. ISSN: 1932-6203. URL: https://doaj.org/article/5728de1040c14d10bcf6112cf0dd0f89.

[22] Isao Fujimoto, Tohru Matsumoto, P. Silva, Masakazu Kobayashi, and Masatake Higashi.
"Mimicking and Evaluating Human Motion to Improve the Imitation Skill of Children with Autism Through a Robot". eng.
In: *International Journal of Social Robotics* 3.4 (2011), pp. 349–357.
ISSN: 1875-4791.

[23] J. Greczek, E. Kaszubski, A. Atrash, and M. Matarić.
"Graded cueing feedback in robot-mediated imitation practice for children with autism spectrum disorders".
In: *The 23rd IEEE International Symposium on Robot and Human Interactive Communication*. 2014, pp. 561–566.
DOI: 10.1109/ROMAN.2014.6926312.

[24] Stephen Haswell Todd.
The turn to the self: A history of autism, 1910-1944. eng. 2015.
URL: http://search.proquest.com/docview/1749781236/.

[25] Robert M. Joseph and James Tanaka.
"Holistic and part-based face recognition in children with autism". In: *Journal of Child Psychology and Psychiatry* 44.4 (2003), pp. 529–542.

DOI: 10.1111/1469-7610.00142. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/1469-7610.00142. URL: https://onlinelibrary.wiley.com/doi/abs/10.1111/1469-7610.00142.

[26] B. A. H. Kargar and M. H. Mahoor. "A pilot study on the eBear socially assistive robot: Implication for interacting with elderly people with moderate depression". In: *2017 IEEE-RAS 17th International Conference on Humanoid Robotics (Humanoids)*. 2017, pp. 756–762.

[27] Roshni Kaushik and Amy Laviers. "Imitation of Human Motion by Low Degree-of-Freedom Simulated Robots and Human Preference for Mappings Driven by Spinal, Arm, and Leg Activity". eng. In: *International Journal of Social Robotics* 11.5 (2019), pp. 765–782. ISSN: 18754791. URL: http://search.proquest.com/docview/2299947592/.

[28] K. Klumper, A. Morbi, K.J Chisholm, R. Beranek, M. Ahmadi, and R. Langlois. "Orientation control of Atlas: A novel motion simulation platform". eng. In: *Mechatronics* 22.8 (2012), pp. 1112–1123. ISSN: 0957-4158.

[29] Alison Knopf. "Autism rates increase slightly: CDC". in: *Brown University Child and Adolescent Behavior Letter* 34.6 (2018), pp. 4–5. ISSN: 1058-1073.

[30] H. Kozima, C. Nakagawa, and Y. Yasuda. "Interactive robots for communication-care: a case-study in autism therapy". eng. In: *ROMAN 2005. IEEE International Workshop on Robot and Human Interactive Communication, 2005.* Vol. 2005. IEEE, 2005, pp. 341–346. ISBN: 0780392744.

[31] Nancy G. Kutner, Rebecca Zhang, Andrew J. Butler, Steven L. Wolf, and Jay L. Alberts. "Quality-of-life change associated with robotic-assisted therapy to improve hand motor function in patients with subacute stroke: a randomized clinical trial." eng. In: *Physical therapy* 90.4 (2010), pp. 493–504. ISSN: 00319023. URL: http://search.proquest.com/docview/733847950/.

[32] Flavio H. B. Lima, Eduardo Poleze, Gabriel P. Das Neves, and Bruno A. Angelico. "Ball Balancing Robot: construction, modeling and control design". eng. In: *2018 13th IEEE International Conference on Industry Applications (INDUSCON)*. IEEE, 2018, pp. 1368–1372. ISBN: 9781538679951.

[33] X. Liu, C. Liu, X. Zhou, X. Zhou, and A. Jiang. "Movement imitation underlying coaching platform for children with ASD". in: *2015 IEEE International Conference on Consumer Electronics - Taiwan*. 2015, pp. 57–58. DOI: 10.1109/ICCE-TW.2015.7217029.

[34] Irina Manouilenko and Susanne Bejerot. "Sukhareva-Prior to Asperger and Kanner". eng. In: *Nordic Journal of Psychiatry* 69.6 (2015), pp. 1761–1764. ISSN: 0803-9488. URL: http://www.tandfonline.com/doi/abs/10.3109/08039488.2015.1005022.

[35] S. Mohammad Mavadati, Huanghao Feng, Anibal Gutierrez, and Mohammad H Mahoor. "Comparing the gaze responses of children with autism and typically developed individuals in human-robot interaction". eng. In: *2014 IEEE-RAS International Conference on Humanoid Robots*. Vol. 2015-. IEEE, 2014, pp. 1128–1133. ISBN: 9781479971749.

[36] Seyedmohammad Mavadati. *Spontaneous Facial Behavior Computing in Human Machine Interaction with Applications in Autism Treatment*. eng. 2015. URL: http://search.proquest.com/docview/1679279504/.

[37] Seyedmohammad Mavadati, Huanghao Feng, Michelle Salvador, Sophia Silver, Anibal Gutierrez, and Mohammad Mahoor. "Robot-based therapeutic protocol for training children with Autism". In: Aug. 2016, pp. 855–860. DOI: 10.1109/ROMAN.2016.7745219.

[38] Micah O Mazurek, Christopher R Engelhardt, and Kelsey E Clark. "Video games from the perspective of adults with autism spectrum disorder". eng. In: *Computers in Human Behavior* 51 (2015), pp. 122–130. ISSN: 0747-5632.

[39] M. Mori, K. F. MacDorman, and N. Kageki.
"The Uncanny Valley [From the Field]".
In: *IEEE Robotics Automation Magazine* 19.2 (2012), pp. 98–100.
ISSN: 1558-223X. DOI: 10.1109/MRA.2012.2192811.

[40] D. N. Nguyen, T. B. Cao, T. H. Pham, N. T. Vo, P. V. Dang, and H. Nam Le. "Design and Control of a Ball-balancing Robot".
In: *2018 4th International Conference on Green Technology and Sustainable Development (GTSD)*. 2018, pp. 317–322.

[41] "Prevalence of Autism Spectrum Disorder among Children in Select Countries Worldwide as of 2020 (per 10,000 Children)." eng.
In: World Population Review (2020). URL:
https://www-statista-com.du.idm.oclc.org/statistics/676354/autism-rate-among-children-select-countries-worldwide/.

[42] Catherine Reed, Paula Beall, Valerie Stone, Lila Kopelioff, Danielle Pulham, and Susan Hepburn.
"Brief Report: Perception of Body Posture—What Individuals With Autism Spectrum Disorder might be Missing". In: *Journal of Autism and Developmental disorders* 37 (Oct. 2007), pp. 1576–84.
DOI: 10.1007/s10803-006-0220-0.

[43] B. Robins, K. Dautenhahn, and J. Dubowski.
"Does appearance matter in the interaction of children with autism with a humanoid robot?" English.
In: *Interaction Studies* 7.3 (2006), pp. 479–512. ISSN: 1572-0373.

[44] James Robinson, J. Holland, M. Hayes, and R. Langlois.
"Velocity-Level Kinematics of the Atlas Spherical Orienting Device Using Omni-Wheels". In: *Transactions of the Canadian Society for Mechanical Engineering* 29 (Dec. 2005).
DOI: 10.1139/tcsme-2005-0046.

[45] M. J. Salvador, S. Silver, and M. H. Mahoor.
"An emotion recognition comparative study of autistic and typically-developing children using the zeno robot". In: *2015 IEEE International Conference on Robotics and Automation (ICRA)*. 2015, pp. 6128–6133.

[46] Michelle Salvador, Anna Marsh, Anibal Gutierrez, and Mohammad Mahoor. "Development of an ABA Autism Intervention Delivered by a Humanoid Robot". In: vol. 9979. Nov. 2016, pp. 551–560. DOI: 10.1007/978-3-319-47437-3_54.

[47] Hans Seywald, Renjith R. Kumar, Carlos M. Roithmayr, Christopher D. Karlgaard, and David M. Bose. *Dynamics and Control of Attitude, Power, and Momentum for a Spacecraft Using Flywheels and Control Moment Gyroscopes - NASA/TP-2003-212178*. eng. Tech. rep. 2003. URL: http://hdl.handle.net/2060/20030038806.

[48] Deborah L. Trout and Howard M. Rosenfeld. "The effect of postural lean and body congruence on the judgment of psychotherapeutic rapport". In: *Journal of Nonverbal Behavior* 4.3 (1980), pp. 176–190. ISSN: 1573-3653. DOI: 10.1007/BF00986818. URL: https://doi.org/10.1007/BF00986818.

[49] Ling-Ling Tsao. "How Much Do We Know about the Importance of Play in Child Development? Review of Research". eng. In: *Childhood Education* 78.4 (2002), pp. 230–33. ISSN: 0009-4056. URL: http://search.proquest.com/docview/62209474/.

[50] Greg Watry. "Designing a Real Life BB-8". eng. In: *R D* (2015). ISSN: 07469179. URL: http://search.proquest.com/docview/1777459882/.

[51] X. Zhang, A. Mollahosseini, A. H. Kargar B., E. Boucher, R. M. Voyles, R. Nielsen, and M. H. Mahoor. "eBear: An expressive Bear-Like robot". In: *The 23rd IEEE International Symposium on Robot and Human Interactive Communication*. 2014, pp. 969–974.